TISHUG NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 10, Number 1 January/February, 1991

Registered by Australia Post - Publication No. NBH5933

1991, the year of software

Asgard Software Page Pro 99

Page Pro Utilities Spell It!

Texaments The Missing Link

CSGD User Disks GIF Mania

MICROdex Artoons CSGD

Harrison Software Music

... to name but a few!

P.O. Box 214, Redfern, New South Wales, Australia, 2016 $3

TIsHUG (Australia) Ltd, based in
Sydney

TIsHUG News Digest

February 1991

All correspondence to:

P.O. Box 214
Redfern, NSW 2016
Australia

The Board

Co-ordinator	
Dick Warburton	(02) 918 8132
Secretary	
Terry Phillips	(02) 797 6313
Treasurer	
Geoff Trott	(042) 29 6629
Directors	
Rolf Schreiber	(042) 84 2980
Russell Welham	(043) 92 4000

Sub-committees

News Digest Editor	
Bob Relyea	(046) 57 1253
BBS Sysop	
Ross Mudie	(02) 456 2122
BBS telephone number	(02) 456 4606
Merchandising	
Percy Harrison	(02) 808 3181
Publications Library	
Warren Welham	(043) 92 4000
Software library	
Rolf Schreiber	(042) 84 2980
Technical co-ordinator	
Lou Amadio	(042) 28 4906

Regional Group Contacts

Central Coast	
Russell Welham	(043) 92 4000
Coffs Harbour	
Kevin Cox	(066) 53 2649
Glebe	
Mike Slattery	(02) 692 0559
Illawarra	
Lou Amadio	(042) 28 4906
Liverpool	
Larry Saunders	(02) 644 7377
Northern Suburbs	
Dennis Norman	(02) 452 3920
Sutherland	
Peter Young	(02) 528 8775

Membership and Subscriptions

Annual Family Dues	$30.00
Associate membership	$10.00
Overseas Airmail Dues	A$60.00
Overseas Surface Mail Dues	A$45.00

TIsHUG Sydney Meeting

The next meeting will start at 2.00 pm on 2nd of February at Ryde Infant School, Tucker Street, Ryde. At 12 pm, before the meeting, there will be a beginners Editor Assembler class for all those interested.

Printed by
The University of Wollongong
Printery

TIsHUG News Digest ISSN 0819-1984

Index

TIsHUG Fairware Author of the Month

The Fairware Author for this month is Martin Smoley for his excellent and informative TI-Base tutorial series. The shop has the ten disks containing these tutorials for sale. All Donations collected at the meeting and sent in will be mailed to him.

Editor's Comment

by Bob Relyea

The Summer holiday was refreshing and I managed to get a bit done around the house as well as on the trusty 'ol computer. I have made progress with that program that I was working on for Michael Ball and I managed to get a bit ahead with the TND editing so I will not be pressured for articles at the beginning of the year at least. There are some good series coming up so I am sure that you will enjoy reading your magazine this coming year, during which we complete our 10th year as a club. Looking back over the past year I can say that I made progress with the computer in my ability to do things with it and in equipment purchase. I hope to be able to do as much if not more this year. Anyway it will be nice to see you all back on the 2nd of February and I am looking forward to a great year. o

Co-ordinator's Report

by Dick Warburton

I have been reading an article about how we use home computers, and it set me thinking about how each one of us will use our computer in 1991. What do you do with your 99/4A? I am sure that different members use them in different ways. Some of us may only have a limited grasp of what can be achieved with a home computer. I think computers are the greatest invention of the electronic age. Without leaving our homes, we can have exotic adventures in the secret passages of the pyramids, play an amazing range of arcade games, challenge our minds to solve puzzles, or simply be entertained in a great variety of ways.

However this is just the beginning, because as we acquire knowledge and skill with them, other usages open up to us. In fact the number of things we can learn to do seems only limited by our imagination, our intelligence, and our motivation. We can learn to program to satisfy our needs, in a variety of languages. We can be creative in art or music, with tools we could hardly have dreamed of 20 years ago. We can learn how things actually work inside the computer, and acquire a working knowledge of electronics and integrated circuits. If all else fails, we can educate ourselves, in a wide variety of disciplines and skills. I have not yet mentioned our use of the machine as a word processor, a data base, or as a business machine. I have a passing interest in the neddies, and I have saved myself considerable financial loss, by keeping a data base of horses' performances and race results, and then checking my ideas by writing a program to evaluate my selections. Let me say that I will never make my fortune punting, but I used to think I could until I used my trusty TI to test my ideas.

I find it frustrating that there are so many other things to do, that interfere with my time on the computer. There is so much I want to learn to do this year. On the hardware scene, I expect to have a really useful set of eproms added to my ramdisk, which will extend its usefulness. I want to learn to be proficient at fixing consoles. I also would like to learn to program effectively in XB and learn some assembly language. I want to write some effective educational programs. There are so many things to do this coming year. The more I learn, the more there seems to be to learn.

I do not know how people who keep buying the latest technology can keep up with the pace of change. It must cost a fortune and be very time consuming to learn to use the latest in computer equipment effectively. There is so much which can be done on a humble TI. I know that I find my TI is a very satisfying machine to use. It does what I tell it, sometimes tempermentally, but I can almost always fix it, or talk to someone who can help. 1991 can be a great year for you if you make the effort to extend your knowledge and proficiency of the TI. If you have a need or a special interest, tell us so that we can try to meet it at the meetings. We do not need to spend lots of money to enjoy our computer, particularly in a time of recession. The TI is a cheap machine to both own and run. It has never been easier or cheaper to expand your machine. Software is cheap. Advice in the club is free. The shop can cater for most of our computer needs at low cost. 1991 can be a year of excitement and challenge. It can be real fun. Join me in making 1991 the best year yet for you and your TI.

See you at the next meeting.

Wanted

I need the instruction booklets for the TI Count General Ledger and the TI Count Mailing System. If you can help, please phone (043) 588-450 and ask for Keith.

Secretary's Notebook

by Terry Phillips

I trust that all members had an enjoyable Christmas and that 1991 brings you everything you wish.

Another year of activities will get under way at the February meeting to be held on Saturday the 2nd. David Sullivan will be the organisor of this years meetings, and I am sure he will come up with some interesting topics. If you have an idea for a meeting format I am sure he would appreciate hearing from you.

The AGM went off well with 38 members in attendance. There were no major controversies and, as I previously advised all members, only the 5 nominations were received for the office of Director. Consequently the Chairman declared Messrs Phillips, Schreiber, Trott, Warburton and Welham duly elected. That dreaded question of incorporation/unincorporation did again raise its ugly head, however it was quickly put to rest with a motion from the floor. At the conclusion of the AGM an enjoyable afternoon tea was served.

As I mentioned in my December column there was a glowing article on our club in the October issue of Micropendium. Well that article is still bearing fruit and a further 6 overseas members have joined us after reading the article. It is a big welcome to the following:

Eunice SPOONER from Waterville, Maine. Eunice had an article on Logo published in the same October issue of Micropendium. If anyone would like to correspond with her on Logo let me know.

Patrick SULLIVAN from Fall River, Massachusetts.

William DUNN from Altamonte Springs, Florida.

George DOWDELL from Chicago, Illinois.

Morris MORGRET from Twin Falls, Idaho.

Michael POSKITT from Wickford, Essex. Michael becomes our second member in England.

In addition we have 2 local members to welcome.

Dick COOPER from Randwick, and Jens BUCHE from Doncaster East (Vic.).

Let's hope that memberships keep rolling in at this level throughout the year, for the bad news is that of 7 membership renewal reminders sent to members with membership expiry dates of 31/12/90, only 2 responses were received.

Over the past 6 weeks or so a good number of local and overseas newsletters have been received. There are some very good articles in these so why not check them out in the library.

I have also received a number of disks which have been passed on to Rolf. Check out his software column for news.

That is all for this month. Hope to see you at the February meeting.

For Sale

A TI-74 programmable computer, complete with a Learn BASIC handbook, a Maths Library module and handbook, a Statistics module and a cassette interface cable. All this for $200.00 ono. For details phone (043) 588-450 and ask for Keith.

TI99/4A console, Mini Expansion, heaps of games, RS232 and much more. Please contact Paul on (043) 591-532.

TIsHUG Shop with Percy Harrison

First I would like to wish you all a belated Happy New Year and apologise for omitting to wish you a Merry Christmas but as I write the shop article at least a month before publication I did not realise that my last one was the Christmas issue.

Also I would like to thank all those members who supported the shop throughout 1990 as it is your support that keeps our head above water and gives us the confidence in the future of the club.

I hope that throughout 1990 you have seen an improvement in our services to our members. We have increased the range of both software and hardware with the addition of 3.5 Drives, 5.25 DSDD Half Height Drives, coloured Monitors, Disk Controllers and various "Do it yourself kits" as well as a Console exchange service and we have now made available a selection of Commercial Software at very reasonable prices. With your continued support 1991 will see an even greater increase in the goodies available with items such as Eprom Ramdisks, Rambo Board, etc, and we hope to see an increase in attendance and activity at our project meetings held every third monday of the month at Ryde Infants school.

The 3.5 and 5.25 drives have been selling exceedingly well and current stocks are getting low so if you require either of these please get your order in promptly as we do not know how much longer our supplies will last.

We have a number of back issue Micropendiums in stock, from 1986 to 1989, which we have now reduced to $0.50 per copy plus packaging and postage. They are not in complete sets and some are in very short supply so if at all possible you should check availability with me before sending in your mail order. I have reasonable stocks of the following issues:

Vol 3 Nos 7,9,10,11
Vol 4 Nos 1 to 11
Vol 5 Nos 2,4,5,6,7,12
Vol 6 Nos 5,6,7,8,9

The average weight of a magazine is 75 grams so please check with your local post office as to the approximate cost of postage for the number of magazines you are ordering as this will vary according to your distance from Sydney, and include this amount with your remittance.

This month I am not including a list of products available from the shop. Please refer to the December issue, page 4 for what is available. However, I have included below an up-to-date list of packaging and posting charges:

	Surface	Airmail
Up to 2 Disks ----------	$1.80	$2.30
From 3 to 9 disks ------	$2.40	$2.70
From 10 to 15 disks ---	$3.10	$4.30
TI Artist Plus ---------	$2.40	$2.70
Display Master ---------	$2.40	$2.70
TI Base ----------------	$2.40	$2.70
TI Sort ----------------	$2.40	$2.70
5.25 inch half-height drives (1.25 kg) ------	refer to your local post office	

My very best wishes for a happy and prosperous New Year to you all and hope to see many of you at our next meeting.

Kits now available

Assembly Squeezing part 5

by Tony McGovern

Very often individual bits in a byte or word are used to flag some condition. The 9900 status register does this all the time and the various conditional jump instructions operate on various combinations of these. Now if you are writing your own code you usually end up using a whole word to store each Boolean flag that in principle needs only one bit, just because it costs more to pack the flags one bit at a time than it saves. There are some intermediate cases however. Mostly you will find these when dealing with system routines and the GPL status byte at >837C. Remember that in GPL the poor old 9900 is weighed down by the task of having to emulate in software an imaginary 8-bit processor. Another situation is exemplified by the directory (SD and QD) routines in Funnelweb. Here the full directory information for up to 127 files must be held in memory (VDP in this case) and yet compete as little as possible for machine resources with other programs. To do this each directory entry is encoded into 14 bytes, which means a certain amount of bitpicking to unravel each entry for display. With 127 entries allowed for, there is scope for the trade-off with the extra bit-picking code. No doubt you will have your own favourite problem.

Suppose you want to check the equals bit in the GPL status register, say after a keyscan or DSRlink call. Your options are a little bit confined here as most of the bitpicking operations assume one or both operands are in a register. You have here to concentrate attention on a particular bit. One way is to set all other bits to zero and the examine the byte (or word or register) to see if it is null. The most general instruction for this is SZCB (or less commonly SZC). You also have to have a mask byte as data somewhere

```
       SZCB @BDF,@GPLST
       JNE  NEWKEY
```

where BDF is the label of a byte with only bit 2 zero, >DF being the appropriate value. This takes 4 words plus a data byte somewhere. An interesting variation on this that can save space if you you have your own BLWP @KSCAN called from several places is to set this flag before leaving the routine.

```
      KSCAN  DATA KWKSP,KSCN1

KSCN1  LIMI 2
       LIMI 0
       LWPI GPLWS
       MOV  R11,@KWKSP+22
       BL   @>E
       LWPI KWKSP
       MOV  R11,@GPLWS+22
       SZCB @BDF,@GPLST
       STST R15
       RTWP
```

The LIMI instructions are to enable interrupts every keyscan and may be omitted if this is not desired. The next few instructions call the console keyscan routine. Saving and restoring R11 of the GPLWS is usually advisable, but not always necessary. The SZCB sets the CPU status bit as before and STST reads the status into R15 just so RTWP can put it back into the CPU status again. Now on every call to KSCAN it need only be followed by a JEQ or JNE to test the GPL status for a repeated key as in

```
       BLWP @KSCAN
       JNE  NEWKEY
```

If this level of byte saving is not called for, a small saving can be made by doing something like

```
       MOVB @GPLST,R0
       SLA  R0,3
       JOC  NEWKEY
```

continued on page 4

TIsHUG Software Column

by Rolf Schreiber

Software Releases for February 1991

The software releases for this month include the latest version of TI Print Shop, the next TI Base Tutorial disk from Martin Smoley and a demonstration disk of a great new game from Asgard Software.

DISK A443 is the sixth disk in a series of nine TI Base tutorial disks, containing the text for lessons #13 and #14, as well as all the associated coomand files and databases mentioned in those articles. The disk is SSSD in format and contains 29 files.

DISK A448 is the latest version, V1.7, of the TIPS program disk, which was written up in the December 1990 issue of the TND. Please refer to that issue for details.

DISK A449 is a demonstration disk of an exciting new game from Asgard Software called Rock Runner. This game is an arcade style game with fascinating graphics, excellent game-play and wonderful music and sound effects. The game has 16 screens, with each level more difficult than the last and requiring a different game strategy. Asgard regarded it as their game of the year in 1990.

Tigercub Software Releases

Jim Peterson was kind enough to send me some disks during the Christmas break, including four disks of the latest TIPS graphics. I will release these shortly, as soon as I get them sorted out. This month I will be releasing three disks from Jim Peterson's public domain collection.

TC-1211, TC-1212 and TC-1213 are all SSSD disks, each containing 28 pictures in TI-Artist format by Ron Griffin. Most of the pictures are of aircraft and cars and the art work is of a very high standard.

Commercial Software Releases

Late last year I wrote to both Asgard and Texaments about coming to some arrangement about distributing their software to our members. Both these companies are happy to give us dealership status, which means that the more we buy, the cheaper each item of software works out for everyone. However, for this arrangement to work satisfactorily, it is necessary for members to:

a) buy the software from the Club in the first place.

b) not give away any illegal (ie "pirated") copies to their friends, but encourage them to purchase their own copy from the Club.

Our prices will be much cheaper than importing the softare yourself, since we are able to get a 20% to 30% discount, depending on the quantity we buy, and the associated costs of importing the software work out cheaper per item. I am hopeful that members will do the right thing and help us to make our venture into commercial software distribution a great sucess. This should result in a steady stream of new software being available through the shop.

At the time of writing to Asgard and Texaments I also ordered some software from them, which only arrived quite recently. I have been kept busy cataloguing it and familiarizing myself with all these new programs. This software will be available through the Shop, so please ask, or write to, Percy to find out what he has in stock. At this stage the shop prices for these items are not yet finalized, but should be settled by the February meeting. At this stage expect them to be only about 5% to 15% greater than the US prices. I intend to order an even larger variety of new commercial software in the near future, so if you have any particular requests, please let us know about them. The following items are currently in stock and should be available for sale at the next meeting:

Character Set & Graphics Design (CSGD) I
Character Set & Graphics Design (CSGD) II
The Missing Link
GIF Mania
Artoons
Page Pro V1.6
Page Pro Utilities
Page Pro Templates
Spell It! (DSDD version)

At this stage I will not have sufficient room to describe all the items in detail, so here goes. The Page Pro Templates, volumes #1 to #12 contain over 80 different Page Pro 99 pages with all associated art work that can easily be customised and printed for professional results. Each volume comprises up to 3 SSSD disks:

Volume #1: Birthday Cards 1, on 2 disks containing 5 cards and 23 pictures.
Volume #2: Birthday Cards 2, on 2 disks containing 8 cards and 40 pictures.
Volume #3: Get Well Cards 1, on 2 disks containing 5 cards and 22 pictures.
Volume #4: Get Well Cards 2, on 2 disks containing 8 cards and 41 pictures.
Volume #5: Assorted Cards 1, on 2 disks containing 8 cards and 38 pictures.
Volume #6: Assorted Cards 2, on 2 disks containing 8 cards and 24 pictures.
Volume #7: Assorted Cards 3, on 1 disk containing 6 cards and 19 pictures.
Volume #8: Invitations 1, on 2 disks containing 8 cards and 30 pictures.
Volume #9: Envelopes 1, on 1 disk containing 3 pages, 12 fonts and 18 pictures.
Volume #10: Assorted Cards 4, on 3 disks containing 8 cards and 24 pictures.
Volume #11: Invitations 2, on 3 disks containing 8 cards and 24 pictures.
Volume #12: Christmas Cards 1, on 3 disks containing 8 cards and 25 pictures.

continued from page 3

As each bit is shifted out it is placed in the Carry status bit, and can be tested with a JOC instruction. This saves a word over checking the bit with

```
MOVB @GPLST,RO
ANDI RO,>2000
JNE  NEWKEY
```

but this last code has the advantage that does an implicit CLR of the register if the jump is not taken. It all very much depends on the details just what is best to do.

continued from page 19

If you want to send a donation to any author outside of the country, remember that bank charges, etc. on a check will take most of what you send. The American dollar bill, as worthless as it has become in this country, is still easily spendable or convertible almost anywhere in the world.

Treasurer's Report

by Geoff Trott

It has been a quiet couple of months on the financial front for the last two months. We should see some more activity from now on.

Income for November	$1399.50
Payments in November	$1524.08
Excess of expenses over income for November	$124.58
Income for December	$1522.26
Payments in December	$ 837.31
Excess of income over expenses for December	$684.95

Techo Time

by Lou Amadio

The Summer holidays can bring a welcome break from work or anything else which tends to take up valuable recreation time! Computing, however, in some form or other, seems to be always with us.

RAMdisk Disable Switch

Anyone working with a RAMdisk will, at some time or other, experience the frustration of having a corrupted ROS lockup up your system, usually at switch on. Although ways of fixing the resulting lockup have been published in the TND previously, sometimes the only way to regain control is to remove the RAMdisk from the PEB. The batteries must also be removed to interrupt the supply to U11 and thus lose the corrupted ROS. The batteries are then re-installed, the RAMdisk is returned to the PEB, the ROS and user files are reloaded from floppy disk. This all takes time, particularly when all you wanted to do was to type out a quick letter.

When I mentioned the problem to Geoff Trott, he suggested that a RAMdisk disable switch might be handy. This is accomplished by switching pin 15 of U9 (74LS259) to ground. I mounted a slide switch at the rear of the PCB in a convenient position so that I can easily reach it with the PEB lid on. Now, if a lockup occurs, I simply activate the switch and I have immediate use of the rest of my system.

Reloading the ROS is accomplished by powering up the system prior to activating the RAMdisk via the above switch, and proceeding as per normal. All in all a simple and very useful suggestion. Thanks Geoff.

Stand Alone Disk System

Just prior to Christmas I built a stand alone disk system for my nephew. He was getting tired of the old TI modules and was pressuring his mother for a games computer. I pointed out that the cost of buying the games machine was only the start and the cost of commercial software was very high and would soon overtake the cost of the hardware.

I suggested that expanding the old TI99/4A with a disk drive might be the way to go. TI PEBs are still expensive and in short supply. Besides, the sheer size tends to put some people off so we settled on a design based on the Direct I/O Interface described previously in the TND.

I managed to find a slim-line drive which was a good colour match for the console and built a "stripped-down" version of the I/O Interface using only one edge connector, a TI disk controller and an outboard power supply. The drive and controller were housed in an compact anodised aluminium box (about 10 mm higher than the console). Look for a photo in the TND when Rolf manages to find some free space.

32K memory expansion, installed in the console, completed the project, which cost about $200 in all.

Expanding a RAMdisk with EPROMs

The other project which took up quite a lot of my (and Geoff's) time was how to put into practice the system devised by Craig Sheehan on using EPROMs to store program files on a RAMdisk.

EPROMs have two main advantages over static RAM chips when it comes to electronic file storage: firstly, they cannot be corrupted or lose their contents; and secondly, they are much cheaper. The main disadvantage is that EPROMs are not readily changeable if one wishes to add, delete or modify files.

Modifying a RAMdisk to use EPROMs requires a fair amount of wiring. The modifications are, however, no more difficult than wiring a HRD+ RAMdisk using a standard Horizon PCB.

Please read the two articles by Craig Sheehan in the May and September 1990 issues of the TND prior to starting this project. Craig's construction article is generic in that it allows any number of EPROMs (within limits) to be added depending on the available RAMdisk partitions.

The project described in this article relates to a specific RAMdisk size and configuration which I believe to be fairly common: ie 384K using 12 x 32K static RAMs (62256) in one layer using the original PCB (designed for 8K chips).

The modifications in this article will allow you to expand your 384K RAMdisk up to 1Mbyte for about $100. Of course you can choose less than this, in 64K increments, for a total EPROM storage of 640K over and above the RAM already fitted.

For convenience, and to allow re-programming if necessary, I used 28 pin IC sockets for all of the EPROM sites.

Choosing EPROM Software

First of all decide on the software to be installed. This will probably involve placing an order with the club shop for the EPROM software that you want. Some software will already be available on EPROMS, eg I have installed Funnelweb, TI Artist Plus, Telco, TI-Base, Multiplan, Picasso and Graphx plus a number of other utility programs that I use. Remember that you must own the software that you wish to have on EPROMs.

EPROM software must be pre-configured to run from either a specific drive number or disk name. If you are using ROS 7.X, then the EPROM files would normally be configured to run from DSK6. Each set of EPROMs is then toggled from the RAMdisk Menu prior to running the desired program. If you are using ROS 8.X then you have a choice of 30 drives (6 to 9, A to Z) and each set of EPROMS can have its own drive identity.

Since each 27C512 EPROM can store 64Kbytes (256 sectors) of data, it is necessary to arrange the files to fully utilize the EPROM capacity. After culling out all unnecessary files (documentation, configuration, etc) I was able to fit the following applications on paired sets of EPROMs:

EPROM #	Software	Sectors
1 and 2	Funnelweb 4.31	497
3 and 4	TI-Artist Plus	508
5 and 6	Picasso + Graphx	510
7 and 8	Telco + TI-Base	505
9 and 10	M/Plan + Utility	498

In addition to the EPROMs you will need some special software (written by Craig) which will tell the ROS that you have EPROMs installed on your RAMdisk.

Wiring the RAMdisk For EPROMs

Having purchased the pre-programmed EPROMs of your choice, the following additional parts will be required to complete the project:

28 pin IC sockets - 1 per EPROM.

74LS08 ICs - 1 per 4 EPROMs.

74LS154 IC - if >2 EPROMs required.

Sockets for the EPROMs are added, in the following order, over the existing 62256 RAM chips: U12, U3, U13, U4, U14, U5, U15, U6, U16, U7. Add the number of sockets corresponding to the number of EPROMs that you want to install. Prior to soldering the sockets onto the RAM chips, bend out pins 20, 27 and 28. These are wired separately to other chips. If the original RAM chips are installed in their own sockets, it will be much easier if you remove them prior to soldering the EPROM sockets on top. Take care with static electricity when handling static RAMs. Orient the sockets in the same direction as the RAM chips. The pins on the sockets that I used required pre-tinning prior to soldering to the RAM chips. Replace the RAM chips in their original sockets.

1) Connect a piece of hookup wire from pin 28 of each EPROM socket to the +5V of any non RAM chip (eg, the RHS of the capacitors above U18 to U23). The EPROMs must not be connected directly to the battery circuit.

2) Bend out pins 1 to 6 and 8 to 13 of a 74LS08 IC. Cut the narrow part of the pins off (except pins 7 and 14). Carefully insert the chip into the U10 position on the PCB and solder pins 7 and 14. This will be U10A. U10B and U10C are bent and cut in the same manner. Do not add U10B until U10A is fully connected. Make the following connections corresponding to the number of added EPROMs (E1 to E10):

EPROM	Pin/Chip	Connect to Pin/Chip
E1	1 of U10A	13 of EXJ, 27 of E1
	2 of U10A	14 of EXJ
	3 of U10A	20 of E1
E2	4 of U10A	15 of EXJ, 27 of E2
	5 of U10A	16 of EXJ
	6 of U10A	20 of E2
E3	8 of U10A	20 of E3
	9 of U10A	1 of U2B, 27 of E3
	10 of U10A	2 of U2B
E4	11 of U10A	20 of E4
	12 of U10A	3 of U2B, 27 of E4
	13 of U10A	4 of U2B
E5	1 of U10B	5 of U2B, 27 of E5
	2 of U10B	6 of U2B
	3 of U10B	20 of E5
E6	4 of U10B	7 of U2B, 27 of E6
	5 of U10B	8 of U2B
	6 of U10B	20 of E6
E7	8 of U10B	20 OF E7
	9 of U10B	9 of U2B, 27 of E7
	10 of U10B	10 of U2B
E8	11 of U10B	20 of E8
	12 of U10B	11 of U2B, 27 of E8
	13 of U10B	13 of U2B
E9	1 of U10C	14 of U2B, 27 of E9
	2 of U10C	15 of U2B
	3 of U10C	20 of E9
E10	4 of U10C	16 of U2B, 27 of E1
	5 of U10C	17 of U2B
	6 of U10C	20 of E10

Pins 8, 9, 10, 11, 12 and 13 of U10C are not used.

Note: If adding more than 2 EPROMs, you need a second 74LS154 chip. Bend out and cut the following pins of U2B: 1 to 11, and 13 to 18. Pre-tin all IC pin stubs to assist in connecting hookup wire. Place the chip over the existing 74LS154 (U2A) and solder all remaining pins (12, 19 to 24). Some of the existing wires to U2A may have to be temporarily removed. Connect a wire from pin 18 of U2B to pin 14 of U20C (top 74LS138) as per page 10 of HRD+ manual.

After all of the wiring has been completed, check it again for errors. Place the card in the PEB (no EPROMs fitted) and verify that it can be used as a normal RAMdisk by loading the ROS and Menu. If all is well, remove the card (after powering down for 1 minute) insert the EPROMs in the correct order (this information should be supplied when you purchase the EPROMs), re-insert the card into the PEB, reload the ROS, configure the RAM and finally run the ESI program to install the EPROMs (see installation instructions by Craig Sheehan on page 8 of the September 1990 issue of the TND.)

After configuring the RAM and EPROM portions of your RAMdisk, save the ROS to a floppy disk (using the CFG program). This will ensure a quick and easy recovery should the ROS ever need to be reloaded.

Note: A different ESI program is required for the 8.X ROS.

WordWriter+ Review

by Ashley Tilling, England

How many of us have come to the end of another evening of cheerfully going through those familiar cassettes of games and half-finished programs by gazing fondly at our unexpanded system (even that phrase left us feeling somewhat emasculated) and thought - where do I go from here? Should the faithful, old TI be abandoned in favour of an Atari, a Mac, an Apricot - or maybe just one of those nice and cheap Amstrads? Or should I continue with the orphaned machine by splashing out vast sums on a PEB and all the associated items? Either way I would be able to move into the word processing league that I had been promising myself for all those years and have something tangible to show from the time spent tapping away in the spare room!

Being the financially cautious sort (if this were not true I would not have a TI!) I decided first to consult Francis 'Parco' Parrish to ask for advice on the costs on expanding the TI. The long and the short of my investigations were that a couple of years ago my interest in the TI was re-kindled by buying a module that Parco called MINI-WRITER III, but is labelled and documented as WORDWRITER+.

For around £80 I received the module (of the usual cartridge type), which has a built-in output port, and a made-up cable to connect directly to the parallel input of a printer. It is, in fact, made by DataBioTics Inc. (c)1987 with a program written by Todd Kaplan. Not having seen TI WRITER, the word processing capabilities of this device did nevertheless seem pretty close to the TIW text editor facilities I had read about.

The first thrill is to enter the world of the 40 column text mode. By using the 40x24 row windowing technique you can create your documents up to 80 columns wide.

On first entering the program you are presented with a prompt line consisting of:

Edit Tabs Quit Files SearcH Lines Purge

with the next line as the Command line for your abbreviated choice. To start on your new page you may change the preset tabs and left/right margins, otherwise simply press E for Edit. The arrow keys are 'live' to move the cursor around the screen (i.e. function E,S,D,X) and the window is moved around the text by using CTRL E or FCTN 6 (window up), CTRL X or FCTN 4 (window down), CTRL D or FCTN 5 (window right) and CTRL S or FCTN 5 (twice - window left).

Text can be inserted using fixed or word-wrap (CTRL 0), and the text closed up after an insertion by reformatting either relative to the left margin (CTRL 2) or relative to the cursor with CTRL R. FCTN 8 inserts a blank line whilst CTRL 8 starts a new paragraph by placing a Carriage return and a Line Feed on the next line. Inevitably characters, lines and blocks of text can also be deleted.

continued on page 16

Assembly Class

Linking Extended BASIC to Assembly

18 LINK IT 18

with Ross Mudie

HOW TO USE THE LINKING UTILITIES FROM EXTENDED BASIC

This tutorial is written chiefly from the material presented at my TISHUG Assembly class in the last quarter of 1990.

When using assembly programs in conjunction with an extended basic program the full resources of the computer are available to the programmer by linking into assembly language program modules. The Editor/ Assembler manual contains much of the information needed by the programmer, but this information is very disjointed because the extended basic information was obviously included as an afterthought. The E/A manual documents linking from TI BASIC in a thorough manner, but there are differences when using Extended Basic.

MEMORY USAGE in EXTENDED BASIC with 32K MEMORY

The extended basic program and the NUMERIC variables are placed in the 24K segment of the 32K memory expansion from >A000 to >FFFF. The actual program lines start at >FFE7 and work down towards >A040. When RUN is executed the pre-scan routine builds a table of variable names in the VDP RAM, with the NUMERIC variables allocated 8 bytes each in fixed positions for that run of the program. To find how much free space exists in the 24K section of the CPU RAM, execute SIZE in the immediate mode. The PROGRAM space free before the program is run or after any modification to the program will show the size of the program. Actual program size is 24488 minus the value from SIZE. The value from SIZE after the program is run will give the amount of memory used by the NUMERIC variables. Subtract the "before run" program space SIZE from the "after run" program space SIZE to find the amount of 24K CPU RAM used by the NUMERIC variables.

The variable "look-up" tables and the STRING variables are located in VDP RAM. As new contents are placed in a string variable, the old location of the string is discarded and the new information is placed in the next available free space in the VDP RAM. Progressively the VDP RAM fills with redundant, discarded strings. A "GARBAGE COLLECTION" then moves all the valid strings up to the top of the VDP RAM, below the variable look- up table. If one or more files are opened, such as disk or RS232, the peripheral access blocks are also allocated space in the VDP RAM. SIZE is also used to find the amount of free space in the VDP RAM. The pre- run value of STACK space FREE shows the amount of VDP RAM available and after the program has been run by executing SIZE again shows how much memory space is used by the variable tables and the STRING variables. If you need to force a "GARBAGE COLLECTION" when it does not matter, rather than when the computer has to, try the little COMPCT program later in this tutorial.

The "standard" linked assembly space is in the LOW CPU RAM, in the 8K block from >2000 to >3FFF. When CALL INIT is executed the assembly support routines are copied from the Extended Basic cartridge into the LOW RAM from >2000 to >24F3. The "linked" assembly will reside from >24F4 to 3FFF. The link names which are the DEF statements in the assembly program are placed in a DEF table consisting of 8 byte blocks, starting from >3FFF and working down. SIZE does not give the free space for the assembly area. The pointers to the space in the LOW RAM are word size values at >2002 and >2004. From extended basic these pointers may be examined by using CALL PEEK(8194,A,B,C,D). C and D which are at location (decimal) 8196 and 8197, give the address of the start of the DEF table if one exists, otherwise the next address after the LOW RAM. A and B give the address of the lowest free space after the assembly linking (INIT) routines if no object file has been loaded or the next free byte in LOW CPU RAM when an object file has been loaded. These memory address values from CALL PEEK are 2 bytes in decimal giving most significant byte followed by the least significant byte. To calculate an address in decimal just multiply the most significant byte by 256 and add the least significant byte, ie, A*256+B gives the decimal address value from the earlier PEEK of the lowest free byte in the LOW RAM. The size of assembly object file loaded may be found by subtracting the value of A*256+B peeked before loading the assembly object file(s) from the A*256+B peeked after loading the assembly object file(s).

CALL INIT must be used in an Extended Basic program or in the immediate mode before CALL LOAD or CALL LINK. CALL PEEK can be used without CALL INIT. After assembly object files have been loaded, if CALL INIT is used again it will reset the pointers as if the assembly object had not been loaded.

CALL LINK allows the extended basic program to LINK to one or more assembly programs. The DEF name of the assembly program is enclosed in quotes in the CALL LINK as follows: CALL LINK("START") would link to an assembly program which included DEF START. When the assembly routine is finished and if it returns control to extended basic it will return to the next executable statement after the CALL LINK. An optional parameter list may be included after the program name in CALL LINK, eg, CALL LINK("START",A,2,C$,"TEST"). These variables, numbers or strings in the parameter list are known as "arguments". Up to 16 arguments may be included in the parameter list.

In the assembly programming environment there are 5 support utilities to assist with interworking between the extended basic and assembly environments, these are: NUMREF, NUMASG, STRREF, STRASG which are used for parameter passing and ERR which is used to report an error condition.

```
NUMREF gets a numeric value from Extended Basic
NUMASG sends a numeric value to Extended Basic
STRREF gets a string from Extended Basic
STRASG sends a string to Extended Basic
ERR    allows the assembly program to return to
       Extended Basic immediate mode via the error
       reporting routine
```

All of the parameter passing support utilities use registers in the current work space to control the actions of the utility. R0 is used to indicate the element in a variable. If the variable is not an array then it will only have an element zero, thus the value for R0 is zero. R1 is used to point to the argument in the parameter list. A value of 1 in R1 signifies the first argument, 2 is the second argument up to 16.

PASSING NUMERIC VALUES.

When a numeric value is passed between Extended Basic and assembly it is in Radix 100 format (base 100) for full precision. This uses 8 bytes and always uses the Floating Point Accumulator (FAC). The FAC is 8 bytes, starting from memory address >834A. Here is a short program example of how a numeric value is passed from Extended Basic to assembly and then changed from a 8 byte full precision number to a 2 byte, word sized, INTEGER value by using the XMLLNK program with the CFI (Convert Floating Point number to Integer) routine.

```
 CLR  R0   Element zero in array or simple variable
 LI   R1,1            Argument 1 in parameter list
 BLWP @NUMREF   Transfer the value from X/B to FAC
 BLWP @XMLLNK        Link to XMLLNK routine in ROM
 DATA >12B8          Tell XMLLNK to Convert Floating
*                     Point to Integer (CFI) in FAC

 MOV  @>834A,R4 Move the value somewhere to use it
```

When passing a numeric value back to Extended basic the opposite is used. If a 2 byte integer value is to be returned it must first be converted into an 8 byte full precision number. Place the word sized value in the FAC, convert it from an Integer to a Floating Point number (CIF) and then set up the element number, argument number and do it.

```
 MOV  R4,@>834A   Move the value from somewhere to FAC
 BLWP @XMLLNK               Link to XMLLNK routine in ROM
 DATA >20                   Tell XMLLNK to Convert Integer
*                          to Floating Point (CIF) in FAC
 CLR  RO        Element zero in array or simple variable
 LI   R1,1                Argument 1 in parameter list
 BLWP @NUMREF         Transfer the value from FAC to X/B
```

The data for the various routines is available on page 416 of the E/A manual. These start at FADD and go down to GVWITE. To find out how to use these routines refer to XMLLNK (pages 259 to 261) for FADD to FCOMP and CIF. Refer to GPLLNK, page 254 for CNS and page 252 for GETSTR. The undocumented ones that I have found work so far are COMPCT which performs an Extended Basic garbage collection on demand, and SCROLL which scrolls the screen by one line each time executed.

PASSING STRINGS

The passing of strings is accomplished with STRREF and STRASG. RO and R1 work the same way for the element number and argument number. R2 is used to specify the address of a buffer for the string. The first byte of the buffer is used to specify the length of the string. When getting a string from extended basic to assembly, place the maximum permissible length of the string in the first byte (byte zero) of the buffer. If the length is going to be too big, which would over-write the memory past the end of the buffer, then the program will be terminated in an error by STRREF.

```
HFF    BYTE >FF    Assignment of value of 255 to label
       EVEN

BUFFER BSS  256    Reservation of 256 bytes for BUFFER

GETSTR CLR  RO    Element 0 in array or simple variable
       LI   R1,3            Argument 3 in parameter list
       LI   R2,BUFFER    Load R2 with address of BUFFER
       MOVB @HFF,@BUFFER     Place max string length in
*                    byte zero (first byte) of BUFFER
       BLWP @STRREF             Actually do the transfer.
```

The actual length of the transferred string will now be in byte 0 of the BUFFER, followed by the contents of the string. The maximum length of a string in Extended Basic is 255 bytes which was allowed by the above routine.

My tutorial LINK-IT15 gives an example of using STRREF and STRASG (There were omissions in the December TND on this file. See the end of this article for the missing parts. ED).

ERROR REPORTING.

In cases where:

a) An out of limits parameter is passed from Extended Basic, or...
b) Some input or operation of the assembly routine creates a condition where the programmer wishes to end the running program in an an error message...

using the error reporting facilities of Extended Basic.

As an example, if ROW and COLUMN are being transferred from Extended Basic, ROW has limit values of 1 and 24. If an out of range value is passed into an assembly routine which writes to the screen, and error trapping of some sort is not employed, then it may write in an unwanted non-screen area of the VDP RAM with unpredictable results.

The following Extended Basic and linked assembly show an example of passing a value via CALL LINK to an assembly routine which accepts the value as a 8 byte precision number (floating point number), converts it to an integer then tests the number against lower and upper limits. If the value is within the preset limits the assembly routine scrolls the screen up 2 lines, gives an OK message and returns to Extended Basic for another value. If a value which is out of range is found, then the program terminates with an error message via BLWP @ERR.

```
100 ! SAVE DSK1.LOAD
110 CALL CLEAR
120 CALL SCREEN(6)
130 FOR S=0 TO 14 :: CALL COLOR(S,16,1):: NEXT S
140 DISPLAY AT(10,1):"LOADING ASSEMBLY..."
150 CALL INIT
160 CALL LOAD("DSK1.O")
170 INPUT "Enter test value ":V
180 CALL LINK("START",V)
190 GOTO 170
```

```
* S=E  O=O                Source and object file names.
       DEF  START         DEF table entry and LINK name

NUMREF EQU  >200C    Address for routine to get number
XMLLNK EQU  >2018  Address of routine for ROM programs
VSBW   EQU  >2020     Address of VDP Single Byte Write
ERR    EQU  >2034             Address of ERRor routine
FAC    EQU  >834A   Addr of Floating Point Accumulator
STATUS EQU  >837C               Address of status byte
GPLWS  EQU  >83E0           Address of GPL Work Space
ERRBV  EQU  >1E00     "Bad Value" Error message equate

SAVRTN BSS  2             Place to save return address
WS     BSS  32         32 bytes for register work space
OKTEXT TEXT 'Value was OK'
       EVEN

ERROR  LI   RO,ERRBV                 Loads RO with >1E00
       BLWP @ERR           Executes link to X/B error msg

START  MOV  R11,@SAVRTN        Remember how to get home!
       LWPI WS           Load reserved register Work Space

       CLR  RO                              Element zero
       LI   R1,1             First argument in link list
       BLWP @NUMREF                        Get the value
       BLWP @XMLLNK                  Execute ROM routine
       DATA >12B8                       Tell XMLLNK CFI
       MOV  @FAC,R4  Put the value in R4 (in this case)
       CI   R4,1              Is the value less than 1?
       JLT  ERROR       If yes then execute error routine
       CI   R4,24          Is the value greater than 24?
       JGT  ERROR       If yes then execute error routine


       LI   RO,738             Where to display on screen
       LI   R2,OKTEXT             Address of start of text
       LI   R3,12      Number of bytes to write on screen
DILOOP MOVB *R2+,R1                Put a byte of text in R1
       AI   R1,>6000     Add hex 60 to display under X/B
       BLWP 2@VSBW              Write a byte on the screen
       INC  RO                   For next screen location
       DEC  R3          Finished yet? Compares with zero
       JNE  DILOOP            If no, go and do next byte

       BLWP @XMLLNK      Branch to XMLLNK routine in ROM
       DATA >26      Tell XMLLNK to Scroll the screen up
       BLWP @XMLLNK      Branch to XMLLNK routine in ROM
       DATA >26      Tell XMLLNK to Scroll the screen up

       CLR  RO           Text book method of returning
       MOVB RO,@STATUS           to the calling program.
       LWPI GPLWS              Reload the GPL Work Space
       MOV  @SAVRTN,R11           Get the return address
       RT                       Return, executes B  *R11

       END
```

When the value transferred from Extended Basic results in an integer less than 1 or greater than 24, the program will terminate with the Extended Basic error message "BAD VALUE IN xx" where xx is the line number of the Extended Basic program which called the assembly routine. The table of Error Equates for Extended Basic linked assembly is found on pages 417 and 418 of the E/A manual. Try changing the error equate value to another in the table to observe the selection of a different error message.

The CFI routine will round up a non-integer value to the next whole number if the non-integer part is is 0.5 or greater or down if it is less than 0.5. Thus if the value is less than 0.5 or 24.5 or greater then the program will terminate in an error.

FORCING A GARBAGE COLLECTION

Another routine available in the XMLLNK routines under Extended Basic is COMPCT.

Extended Basic CALL LINK("COMPCT")

Assembly..

```
XMLLNK EQU  >2018

       DEF  COMPCT  Entry point name in linked assembly

COMPCT BLWP @XMLLNK                 Use routine in ROM
       DATA 0      Specify which routine, E/A page 416
       RT                    Return to Extended Basic

       END
```

If you have read the whole tutorial this far - congratulations! Now type the Assembly and Extended Basic programs in, assemble with the R option, debug any typographical errors and give it a go, good luck.

We apologise for the fact that part of the article in the December issue regarding Link-It 15 was missing as there were transmission (or dumping?) problems from the BBS to my system and I (Bob) did not pick it. I will not repeat the entire article for lack of space. Everything is OK until the top right hand corner of page 9, line 2. The following is what should have been included between line 1 of the TND and line 4:

```
* string into upper case. It will handle strings up to
* 255 bytes in length. Type in the extended basic
* program to test the utility from the keyboard.
* Ross Mudie 21/12/87

EXTENDED BASIC TEST PROGRAM

* 100  ! SAVE DSK1.LOAD
* 110  CALL INIT              ! LOAD ASSEMBLY UTILITIES
* 120  CALL LOAD("DSK1.UO")   ! LOAD ASSEMBLY OBJECT FILE
* 130  ACCEPT S$              ! INPUT FOR UP TO 255 BYTES
* 140  CALL LINK("CONVUP",S$) !  DO THE CONVERSION ON S$
* 150  DISPLAY S$        ! SHOW THE RESULTS ON THE SCREEN
* 160  GOTO 130               ! GO BACK FOR ANOTHER TRY

       DEF  CONVUP   CALL LINK("CONVUP",S$)

STRREF  EQU  >2014  Utility which gets a string from X/B
STRASG  EQU  >2010   Utility which sends a string to X/B
GPLWS   EQU  >83E0  Address of GPL Work Space (for X/B)
STATUS  EQU  >837C               GPL STATUS byte address

FF      BYTE >FF  Used for max string length in BUFFER+0
BH20    BYTE >20    Used to subtract from lower case to
                                         make upper case

CONVUP  MOV  R11,@SAVRTN          Save the return address
        LWPI WS         Load our own register work space

        CLR  R0              Element zero in variable S$
        LI   R1,1              Argument 1 in link list
        MOVB @FF,@BUFFER  Max string length in Buffer+0
        LI   R2,BUFFER            Where to put the string
        BLWP @STRREF       Get the string from extd basic

        MOVB @BUFFER,R3    Get actual string length byte
        SRL  R3,8         Make byte-sized value into word

        MOV  R3,R3                  Is it a null string?
        JEQ  END       If a null go back to ext'd basic

        LI   R4,>6100            Hex 61 is 'a' lower case
        LI   R5,>7A00            Hex 7A is 'z' lower case

        LI   R6,BUFFER+1        The value in R6 is now the
*                                 address of the first byte
*                               of the string in the BUFFER
```

We trust that this supplies the 'missing link' between the December TND and the intended text. Sorry for any inconvenience.

Why Learn to Program?

by Jim Peterson, Tigercub Software, USA

Why should you learn to program? To make money? No way! If you could write a program to guarantee world peace, eliminate hunger and cure AIDS, you could not make money selling it to the TI world!

Why should you learn to program? To contribute something to the TI world? OK, but do not expect any thanks! Contributing a program to the TI public domain is like dropping a pebble into a bottomless dry well - you will never hear a splash, not even a thud.

Why should you learn to program? Because no one has written the program you need? Well, now you have a good reason! Since there is neither money nor recognition in programming, the programmers tend to write what they feel like writing, not what you want them to write.

Why should you learn to program? For one reason, because I know that you would like to make some changes in the programs that you use frequently. I know that, because the only feedback I ever get is from people who wish that I would change this or that! You really would not have to learn very much to change colours, add or silence a beep or a burp, output to disk instead of printer, etc. etc.

Beyond that, unraveling someone else's code can be tricky and frustrating (and I pity anyone who tries to unravel my code!). Often I find it easier to just rewrite the basic idea in my own way.

If you do modify someone's program, please put a note on the title screen, or at least in a REM, that you did so - and unless you are very sure that you have not introduced a bug, do not distribute your version! Programmers do not like to be blamed for other people's mistakes, and the sales of good programs have been ruined by the bad reputation resulting from pirated, modified and bugged copies.

But, the real reason for learning to program - it is fun, it is challenging, it is creative! There is something very satisfying about getting an idea to make the computer do something it has never done (as far as you know!) and then succeeding in making it do what you want. There is a thrill in pushing the limits of that obsolete tiny TI pea brain just a little bit farther.

There are those who prefer to exercise their creativity with the soldering iron, those who can plug in chips and soup up a Model T computer to run like a Ferrari. I regard them with awe and wonderment, and I am glad they are around. Without them, I would not have my RamDisk, and my equipment would not get repaired.

Personally, I am the ultimate klutz. If I approach my car with a screwdriver, all four tires go flat. My one feeble attempt to repair my P-Box resulted in failure, expense and embarrassment. But, without having more than a faint idea what goes on beneath that keyboard, I have learned to punch the keys (two right fingers and a left thumb) and create hundreds of programs and routines which have given me a great deal of satisfaction.

It has been fun! You should try it sometime.

Subject Index 1990

Title	Description	Author	Page No.
3.5 inch drives	Hardware hints	Takach,Ben	8.12
4-D graphics	Software hints	Peterson,Jim	7.22
4-D graphics	Software hints	Peterson,Jim	8.20
Advertising really works!	General interest	Schreiber,Rolf	11.24
Agenda for 1990 AGM	Club news	Phillips,Terry	1.02
Agenda of 4th AGM	Club news		10.04
Apple peeks	Software hints	Callaghan,Mike	3.17
Assembler titbit	Software hints	Weasel,Inc.	9.03
Assembler tutorial #1	Software hints	McCormick,Mack	10.23
Assembly class	General interest	Mudie,Ross	9.14
	General interest	Mudie,Ross	10.09
	General interest	Mudie,Ross	11.09
Assembly squeezing, part 1	Software hints	McGovern,Tony	7.22
part 2	Software hints	McGovern,Tony	9.17
part 3	Software hints	McGovern,Tony	10.06
part 4	Software hints	McGovern,Tony	11.08
Assembly squish	Software hints	McGovern,Tony	7.21
Assembly tutorials	General interest	Mudie,Ross	8.02
Author index 1989	General interest		1.15
Awards, USA	General interest	Gaskill,Bill	9.19
Bases in c99	Software hints	Pulley,Clint	6.12
Beginning Forth #1	Software hints	Raguse,Earl	11.17
BOOT, bug fix	Software hints	Schreiber,Rolf	7.06
program loader	Software review	Slattery,Mike	5.10
Braille'n speak	Hardware review	Hott,Irwin	11.25
Brain, a review	Software review	Rosenberg,Howie	1.10
Bridge on TI99/4A	Software idea	Gleed,Peter	4.07
Bulletin board users	General interest	Shaw,Stephen	7.09
Burning EPROMs	Software review	Sheehan,Craig	4.04
c99 TIPS slide show	Software review	Murphy,Terrence	11.15
TIPS to TI-Artist convert	Software review	Murphy,Terrence	11.15
TIPS viewer	Software review	Murphy,Terrence	11.15
function library	Software hints		3.27
Cataloguing software library	Software hints	Schreiber,Rolf	11.04
with multiplan	Software hints	Relyea,Bob	7.19
Co-ordinators report	General news	Warburton,Dick	1.02
	General news	Warburton,Dick	2.02
	General news	Warburton,Dick	3.02
	General news	Warburton,Dick	4.02
	General news	Warburton,Dick	5.02
	General news	Warburton,Dick	6.02
	General news	Warburton,Dick	7.02
	General news	Warburton,Dick	8.02
	General news	Warburton,Dick	9.02
	General news	Warburton,Dick	10.02
	General news	Warburton,Dick	11.02
Colour Monitor interface	Hardware project	Amadio,Lou	1.05
	Hardware project	Trott,Geoff	1.05
Colours in Editor Assembler	Software hints	Atkinson,Terry	5.10
Communicators	BBS information	Mudie,Ross	4.13
	BBS Information	Mudie,Ross	6.04
	BBS information	Mudie,Ross	7.09
Computers are innocent!	General interest	Takach,Ben	8.04
Configure printer for DM1000	Software hints	Moore,Fred	11.17
for Fweb 4.10	Software hints	Owen,John	11.24
CorComp disk controller	General interest	Moseid,William	4.27
Correction to tips #58	Software hints	Peterson,Jim	5.05
Cracked	Software hints	Jack,Cracker	6.20
	Software hints	Jack,Cracker	7.20
Customizing Superbug 2	Software hints	Dohmann,Edgar	1.08
Cutthroats part 1	Adventure hints	Scorpia	7.10
part 2	Adventure hints	Scorpia	9.06
part 3	Adventure hints	Scorpia	10.26
DataWriter version 1.2	Software review	Peterson,Jim	5.04
Diagnosing simple HFDC fault	Hardware hints	Takach,Ben	6.10
Disk file formats	Software hints	Mudie,Ross	10.16
Disk peripheral	Specifications		3.21
	Specifications		4.25
Disk reviews	Software review	Shaw,Stephen	7.14
	Software review	Shaw,Stephen	10.17
Editor's comment	General interest	Relyea,Bob	3.01
	General interest	Relyea,Bob	4.01
	General interest	Relyea,Bob	5.01
	General interest	Relyea,Bob	6.01
	General interest	Relyea,Bob	7.01
	General interest	Relyea,Bob	8.01
	General interest	Relyea,Bob	9.01
	General interest	Relyea,Bob	10.01
	General interest	Relyea,Bob	11.01
Editor's reply	General interest	Relyea,Bob	8.03
Education Modules	Software review	Shaw,Stephen	6.15
Emulate file creation	Software hints	Schreiber,Rolf	2.33
Enhanced BASIC	Software hints	Shaw,Stephen	10.11
EPROM RAMdisk	Hardware project	Sheehan,Craig	8.07
Expanding your PEB	Hardware hints	Schreiber,Rolf	9.20
Extended BASIC loaders	Software hints	Mudie,Ross	8.09
tips #1-4	Software hints	Swedlow,Jim	10.12
tips #15	Software hints	Swedlow,Jim	8.13
trace to printer	Software hints	Mudie,Ross	5.18
Extended BASIC tutorial	Software hints	McGovern,Tony	1.09
	Software hints	McGovern,Tony	2.09
	Software hints	McGovern,Tony	3.07
Fixing blown disks	Software hints	Massey,Howard	6.19
Forth to you too!	Software hints		1.24
	Software hints		2.20
	Software hints		3.24
Four-A/talk	Software hints	Gaskill,Bill	3.09
From the bulletin board	Mail to all		1.24
	Mail to all		2.30
	Mail to all		3.22
	Mail to all		4.23
	Mail to all		5.09
	Mail to all		11.10
From the editor	General interest	Relyea,Bob	2.02
From the service bench	AT disk contr,RAMdisk	Trott,Geoff	2.06
	Consoles	Trott,Geoff	6.21
	RAMdisk	Trott,Geoff	3.29
Future of TIsHUG	Club news	Takach,Ben	4.02
G language	Software review	Shaw,Stephen	6.11
Game of NIT-WIT	Software review	Lang,Chris	2.25
Games catalogue	Software review	Peterson,Jim	9.11
Games information	Adventures	Brown,Robert	2.15
	Spellbreaker	Brown,Robert	4.14
	Spellbreaker	Brown,Robert	5.14
Games modules	Software review	Shaw,Stephen	7.13
Getting a line on your program	Software hints	Presser,Joe	1.25
Graphics for Myarc XB	Software hints	Shaw,Stephen	11.20
Greeting cards	Software hints	Ruggeri,Alf	10.13
Handy hints	Software hints		11.18
Hard disk setting up	Software hints	Wilkinson,Derek	7.08
Hardware corner	Hardware hints	Roseen,Richard	11.26
High resolution graphics	Software hints	Shaw,Stephen	3.26
How does it work?	General interest	Trott,Geoff	2.34
	General interest	Trott,Geoff	3.29
	General interest	Trott,Geoff	4.29
	General interest	Trott,Geoff	5.22
Letter to editor	Future of TIsHUG	Amadio,Lou	7.03
	Future of TIsHUG	Beuermann,Tony	8.03
	Incorporation	Amadio,Lou	5.02
	Intruder alarm	Ball,Michael	11.03
	Monitors	Mudie,Ross	6.02
	Program idea	Ball,Michael	8.03
Linkit #15	Software hints	Mudie,Ross	11.09
Living with spiders, part 1	Software hints	McGovern,Tony	7.17
part 2	Software hints	McGovern,Tony	8.15
part 3	Software hints	McGovern,Tony	9.15
MICROpendium review	General interest	Brashear,Harry	11.02
Minutes of 1989 AGM	Club news	Phillips,Terry	1.03
3rd AGM	Club news	Phillips,Terry	10.03
special GM	Club news	Phillips,Terry	10.04
Modified formatter commands	Word processing		2.24
Modular programming	Software hints	Christensen,Garry	10.15
Module reviews	General interest	Shaw,Stephen	4.28
Monitor interface power supply	Hardware hints	Ockenden,Eric	8.07
More PEEKs and POKEs	Software hints	Schreiber,Rolf	2.07
Multi-function card features	Hardware review	Schubert,Peter	6.08
My use for computer	General interest	Beuermann,Tony	11.10
	General interest	Relyea,Bob	11.10
New location for Sydney meetings	Club news	Mudie,Ross	2.12
Newsletter annual report	Club news	Trott,Geoff	2.01
Newsletter update	General interest	Relyea,Bob	3.31
	General interest	Relyea,Bob	5.22
	General interest	Relyea,Bob	8.19
	General interest	Trott,Geoff	4.30
Notice of General Meeting	Club news		6.03
Oasis pensive abacutors	General interest	Bowser,Garry	9.02
Ontplopper!	Software review	Shaw,Stephen	9.17
Ordering library software	Club news	Schreiber,Rolf	5.04
Overcoming alpha-lock	Hardware hints	Paine,John	4.16
Page Pro 99	Software review	Johnson,Ed	1.27
part 3	Software review	Asgard	9.10
utilities	Software review	Scheidemantle,Paul	9.07
PC-clone power supplies	Hardware hints	McGovern,Tony	11.23
Periscope instructions	Game	Pewterware	1.11
Peter Gleed, in memorial	General interest	M,W	10.02
Pix pro	Software review	Reiss,Jim	5.07
Power supplies and transfomers	Hardware hints	McGovern,Tony	8.17
Printer CTRL[U] commands	Word processing	Saunders,Larry	4.21
Program information	BBS information	Mudie,Ross	7.09
Program to type in	Ancient history quiz	Maker,Vincent	9.18
	Australian flag	Longmuir,Bill	4.15
	Chemistry test	Grunge	2.27
	Grue stew		3.15
	Horse sprite		9.18
Programs from TI*MES library	Software review	Shaw,Stephen	3.25
Programs to write programs #6	Software hints	Peterson,Jim	6.16
Project round-up	Hardware review	Amadio,Lou	6.09
Putting it all together #6	Software review	Peterson,Jim	5.18
Quick Run review	Software review	Shaw,Stephen	2.22
RAM memory tester	Assembler code	Hansen,Tor	2.31
Rambles	General interest	Shaw,Stephen	2.13
	General interest	Shaw,Stephen	5.15
	General interest	Shaw,Stephen	6.15
	Software hints	Shaw,Stephen	7.11
	Software hints	Shaw,Stephen	11.11
RAMdisk folly	General interest	Forthwright	3.28
Real programmers, part 1	General interest		8.21
part 2	General interest		9.21
REF/DEF table	Software hints	Rebel,Eric-Paul	2.23
Regional group reports	General interest		1.31
	General interest		2.35
	General interest		3.31
	General interest		4.31
	General interest		5.23
	General interest		6.23
	General interest		7.23
	General interest		8.23
	General interest		9.23
	General interest		10.27
	General interest		11.27
Reviews	General interest	Shaw,Stephen	3.23
	General interest	Shaw,Stephen	4.17

Title	Description	Author	Page No.
SAVE/RECALL user notes	Software hints	Kaplan,Todd	1.23
Secretary's notebook	Club news	Phillips,Terry	1.04
	Club news	Phillips,Terry	2.03
	Club news	Phillips,Terry	3.02
	Club news	Phillips,Terry	4.03
	Club news	Phillips,Terry	5.03
	Club news	Phillips,Terry	6.04
	Club news	Phillips,Terry	7.04
	Club news	Phillips,Terry	8.04
	Club news	Phillips,Terry	9.03
	Club news	Phillips,Terry	10.05
	Club news	Phillips,Terry	11.03
Secretary's report to 6/90	Club news	Phillips,Terry	10.05
Sheehan EPROM disk	Hardware hints	Trott,Geoff	11.08
Small list	Software hints	Rebel,Eric-Paul	3.27
Software extra	Club software	Schreiber,Rolf	2.03
Sorcerer part 1	Adventure hints	Scorpia	2.15
part 2	Adventure hints	Scorpia	3.10
part 3	Adventure hints	Scorpia	4.31
part 4	Adventure hints	Scorpia	5.19
Spellbreaker part 1	Adventure hints	Scorpia	11.12
Subject index 1989	General interest		1.14
Summer storms, computers, fax	General interest	Mudie,Ross	11.09
Super Extended BASIC	Software review	Shaw,Stephen	4.11
Superbug 2, a description	Software review	Dohmann,Edgar	1.08
Symphony	General interest		6.05
Teachers use the TI99/4A	General interest	Relyea,Bob	5.19
Techo time	32K memory expansion	Amadio,Lou	4.08
	AT card, travels	Amadio,Lou	10.08
	Hardware hints	Amadio,Lou	5.06
	Hardware hints	Amadio,Lou	6.07
	Hardware hints	Amadio,Lou	7.07
	Hardware hints	Amadio,Lou	8.06
	Hardware project	Amadio,Lou	1.05
	Multifunction card	Amadio,Lou	3.05
	PIO, drive switch	Amadio,Lou	9.13
	RGB inter,AT disk cont	Amadio,Lou	2.05
	RS232 for AT card	Amadio,Lou	11.07
Testing PC-Transfer	Software review	Takach,Ben	4.12
Tetris, a game	Software review		3.30
Text on disk	Software hints	Shaw,Stephen	10.22
They're off	General interest	Trott,Geoff	1.01
TI Artist plus	Software review	Shaw,Stephen	3.14
TI hacker's lament	General interest	Takach,Ben	8.12
TI learning machine	General interest	Amundsen,M	5.11
TI*mes disk library report	General interest	Shaw,Stephen	6.13
TI-Base 2.0 review	Software review	Lilley,Rick	1.17
files to DV80	Software hints	Relyea,Bob	6.12
TI-Base tutorial	Data base	Smoley,Martin	1.13
	Data base	Smoley,Martin	2.11
	Software review	Saunders,Larry	4.09
TI-Bits #1	Software hints	Swedlow,Jim	10.12
#13 and #15	Software hints	Swedlow,Jim	3.18
#16	Software hints	Swedlow,Jim	8.14
#17	Software hints	Swedlow,Jim	8.14
TI-Faire speech at Melbourne	General interest	Christensen,Garry	4.03
TI-Keys	Software review	Johnson,Wes	4.24
with Funnelweb	Software hints	Good,Charles	4.24

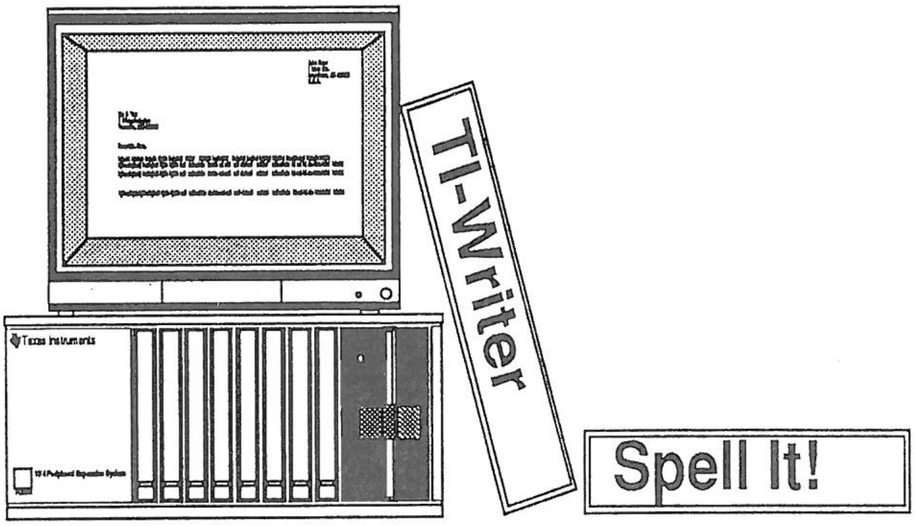
Title	Description	Author	Page No.
TI-Writer hints	Word processing	Relyea,Bob	2.04
once again	Software hints	McGovern,Tony	7.15
version 4.0	Word processing	Green,Art	1.21
TI99/4A user groups	General interest	Peterson,Jim	2.16
TIdBits #2	Software hints	Swedlow,Jim	11.16
TIPS documentation	Software review	Wolcott,Ron	11.13
Tips from the tigercub #38	Software hints	Peterson,Jim	1.19
#39	Software hints	Peterson,Jim	2.21
#40	Software hints	Peterson,Jim	3.19
#41	Software hints	Peterson,Jim	4.19
#59	Software hints	Peterson,Jim	6.17
TIsHUG shop report	Club news	Harrison,Percy	4.05
	Club news	Harrison,Percy	4.05
	Club news	Harrison,Percy	5.05
	Club news	Harrison,Percy	6.05
	Club news	Harrison,Percy	7.06
	Club news	Harrison,Percy	8.05
	Club news	Harrison,Percy	9.04
	Club news	Harrison,Percy	10.06
	Club news	Harrison,Percy	11.04
TIsHUG software column	Club software	Phillips,Terry	1.02
	Club software	Phillips,Terry	2.03
	Club software	Schreiber,Rolf	3.03
	Club software	Schreiber,Rolf	4.06
	Club software	Schreiber,Rolf	5.04
	Club software	Schreiber,Rolf	6.06
	Club software	Schreiber,Rolf	7.05
	Club software	Schreiber,Rolf	8.05
	Club software	Schreiber,Rolf	9.05
	Club software	Schreiber,Rolf	10.07
	Club software	Schreiber,Rolf	11.05
Treasurer's report	Club news	Trott,Geoff	4.07
	Club news	Trott,Geoff	5.03
	Club news	Trott,Geoff	6.06
	Club news	Trott,Geoff	7.08
	Club news	Trott,Geoff	9.17
	Club news	Trott,Geoff	10.21
	Club news	Trott,Geoff	11.17
UK users annual meeting	General interest	Shaw,Stephen	8.22
Upgrading PEB power supply	Hardware hints	Bray,Eric	5.21
Uses for my computer	General interest	Mudie,Ross	10.26
Uses for your computer?	General interest	Mudie,Ross	11.10
Vector graphics and transforms	Software hints	Shaw,Stephen	11.21
Viatel, what is it?	General interest	Nielsen,R	10.10
WIT or WIT-OUT	Software review	Lang,Chris	3.11
XB tips #14	Software hints	Swedlow,Jim	4.13
Younger set	Music programs	Maker,Vincent	11.19
	Program	Maker,Vincent	6.22
	Program	Maker,Vincent	7.14
	Program	Maker,Vincent	10.27
	Program, adventure	Maker,Vincent	2.29
	Program, adventure	Maker,Vincent	3.17
	Program, adventure	Maker,Vincent	5.20
	Program, adventure	Maker,Vincent	9.14

Author Index 1990

Author	Title	Description	Page No.
Amadio,Lou	Colour Monitor interface	Hardware project	1.05
	Letter to editor	Future of TIsHUG	7.03
	Letter to editor	Incorporation	5.02
	Project round-up	Hardware review	6.09
	Techo time	32K memory expansion	4.08
	Techo time	AT card, travels	10.08
	Techo time	Hardware hints	5.06
	Techo time	Hardware hints	6.07
	Techo time	Hardware hints	7.07
	Techo time	Hardware hints	8.06
	Techo time	Hardware project	1.05
	Techo time	Multifunction card	3.05
	Techo time	PIO, drive switch	9.13
	Techo time	RGB inter,AT disk cont.	2.05
	Techo time	RS232 for AT card	11.07
Amundsen,M			
OH,USA	TI learning machine	General interest	5.11
Asgard			
USA	Page Pro 99 part 3	Software review	9.10
Atkinson,Terry			
NS,Canada	Colours in Editor Assembler	Software hints	5.10
Ball,Michael	Letter to editor	Intruder alarm	11.03
	Letter to editor	Program idea	8.03
Beuermann,Tony	Letter to editor	Future of TIsHUG	8.03
	My use for computer	General interest	11.10
Bowser,Garry			
Ont,Canada	Oasis pensive abacutors	General interest	9.02
Brashear,Harry			
USA	MICROpendium review	General interest	11.02
Bray,Eric			
PA,USA	Upgrading PEB power supply	Hardware hints	5.21
Brown,Robert	Games information	Adventures	2.15
	Games information	Spellbreaker	4.14
	Games information	Spellbreaker	5.14
Callaghan,Mike			
USA	Apple peeks	Software hints	3.17
Christensen,Garry	Modular programming	Software hints	10.15
Qld	TI-Faire speech at Melbourne	General interest	4.03
Dohmann,Edgar	Customizing Superbug 2	Software hints	1.08
USA	Superbug 2, a description	Software review	1.08
Forthwright			
Qld	RAMdisk folly	General interest	3.28
Gaskill,Bill	Awards, USA	General interest	9.19
USA	Four-A/talk	Software hints	3.09
Gleed,Peter			
Vic	Bridge on TI99/4A	Software idea	4.07

Author	Title	Description	Page No.
Good,Charles			
USA	TI-Keys with Funnelweb	Software hints	4.24
Green,Art			
Canada	TI-Writer version 4.0	Word processing	1.21
Grunge			
USA	Program to type in	Chemistry test	2.27
Hansen,Tor			
Ont,Canada	RAM memory tester	Assembler code	2.31
Harrison,Percy	TIsHUG shop report	Club news	4.05
	TIsHUG shop report	Club news	4.05
	TIsHUG shop report	Club news	5.05
	TIsHUG shop report	Club news	6.05
	TIsHUG shop report	Club news	7.06
	TIsHUG shop report	Club news	8.05
	TIsHUG shop report	Club news	9.04
	TIsHUG shop report	Club news	10.06
	TIsHUG shop report	Club news	11.04
Hott,Irwin			
OH,USA	Braille'n speak	Hardware review	11.25
Jack,Cracker	Cracked	Software hints	6.20
	Cracked	Software hints	7.20
Johnson,Ed			
USA	Page Pro 99	Software review	1.27
Johnson,Wes			
USA	TI-Keys	Software review	4.24
Kaplan,Todd			
USA	SAVE/RECALL user notes	Software hints	1.23
Lang,Chris	Game of NIT-WIT	Software review	2.25
USA	WIT or WIT-OUT	Software review	3.11
Lilley,Rick			
Vic	TI-Base 2.0 review	Software review	1.17
Longmuir,Bill	Program to type in	Australian flag	4.15
M,W			
Vic	Peter Gleed, in memorial	General interest	10.02
Maker,Vincent	Program to type in	Ancient history quiz	9.18
	Younger set	Music programs	11.19
	Younger set	Program	6.22
	Younger set	Program	7.14
	Younger set	Program	10.27
	Younger set	Program, adventure	2.29
	Younger set	Program, adventure	3.17
	Younger set	Program, adventure	5.20
	Younger set	Program, adventure	9.14
Massey,Howard			
USA	Fixing blown disks	Software hints	6.19

Spell It!



"The Best Thing Next to your Word Processor"

Introducing the best thing next to your word processor, the *Spell It!* spelling checker by Jim Reiss and Asgard Software.

Spell It! brings spell checking on the TI-99/4A and Myarc Geneve 9640 into the '90s with all the features and speed of modern PC and Macintosh spellers. With *Spell It!* you'll never have to worry about embarrassing spelling mistakes again in your *TI-Writer, RAG-Writer, BA-Writer, Funnelweb, My-Word,* or *WordWriter* documents. Compatible with any word processor that can store documents in the TI standard text format (display/variable 80 files), *Spell It!* for the first time makes checking your spelling on the 4A easier than writing the words in the first place.

Among other things, *Spell It!* is many times faster than any other TI spelling checker, it makes corrections directly to your text file so you don't have to return to the editor, lets you view the context the word is used in, allows you to add words as you check text files to a user dictionary limited only by available disk space, has no limit on text file size, lets you scan through the dictionary if you aren't sure of a spelling, and even supports foreign language character sets (é, ü, å, etc.).

Finally, *Spell It!* is available in three versions - SS/SD and DS/DD disk versions with over 25,000 words, and a hard-disk version with over 200,000 words. The proprietary SmartCheck™ algorithm in *Spell It!* makes this equivalent to a dictionary several times as large since *Spell It!* doesn't have to wade through the plural and possessive versions of words often found in other spelling checker dictionaries.

Requires at a minimum 32K, one SS/SD disk drive and TI-Writer or a compatible. Recommended system of at least one DS/DD disk drive. Provided unprotected and RAM-disk compatible. Configurable for 80-columns on the TI-99/4A and Geneve.

DS/DD - $19.95
SS/SD - $24.95
HFDC - $34.95
U.S. Add $2.50 S&H
Canada Add $4.00
Airmail Add $6.00

ASGARD SOFTWARE
P.O. Box 10306 • Rockville, MD 20849 • (703)255-3085

Refer to page 4 for TIsHUG shop prices

TIsHUG (Australia)

	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
February 1991		1	2 TIsHUG meeting Sydney	3	4	5	6 TND deadline	7	8	9	10	11	12	13	14	15	16
March 1991		1	2 TIsHUG meeting Sydney	3	4	5	6 TND deadline	7	8	9	10	11	12	13	14	15	16
April 1991					1 Easter Monday	2 School holidays	3 School holidays	4 School holidays	5 School holidays	6 TIsHUG meeting Sydney	7 School holidays	8	9	10 TND deadline	11	12	13
May 1991							1	2	3	4 TIsHUG meeting Sydney	5	6	7	8 TND deadline	9	10	11
			Sat	Sun	Mon	Tue											
June 1991			1 TIsHUG meeting Sydney	2	3	4	5 TND deadline	6	7	8	9	10 Queen's birthday	11	12	13	14	15
July 1991					1 School holidays	2 School holidays	3 School holidays	4 School holidays	5 School holidays	6 TIsHUG meeting Sydney	7 School holidays	8 School holidays	9 School holidays	10 TND deadline School holidays	11 School holidays	12 School holidays	13 School holidays
	Thu	Fri	Sat	Sun													
August 1991	1	2	3 TIsHUG meeting Sydney	4	5 Bank holiday	6	7 TND deadline	8	9	10	11	12	13	14	15	16	17
September 1991				1	2	3	4	5	6	7 TIsHUG meeting Sydney	8	9	10	11 TND deadline	12	13	14
October 1991						1 School holidays	2 School holidays	3 School holidays	4 School holidays	5 TIsHUG meeting Sydney	6 School holidays	7 Labour Day	8 School holidays	9 TND deadline School holidays	10 School holidays	11 School holidays	12 School holidays
		Fri	Sat	Sun	Mon												
November 1991		1	2 TIsHUG meeting Sydney	3	4	5	6 TND deadline	7	8	9	10	11	12	13	14	15	16
December 1991				1	2	3	4	5	6	7 TIsHUG Annual General meeting	8	9	10	11	12	13	14 School holidays
January 1992							1 New Year's Day	2 School holidays	3 School holidays	4 School holidays	5 School holidays	6 School holidays	7 School holidays	8 TND deadline School holidays	9 School holidays	10 School holidays	11 School holidays

lanning Calendar, 1991 - 1992

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri
	18	19	20	21	22	23	24	25	26	27	28								
	18	19	20	21	22	23	24	25	26	27	28	29 Good Friday	30 School holidays	31 Easter Day					
	15	16	17	18	19	20	21	22	23	24	25 Anzac Day	26	27	28	29	30			
	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	17	18	19	20	21	22	23	24	25	26	27	28	29 School holidays	30 School holidays					
ol ys	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31		
	19	20	21	22	23	24	25	26	27	28	29	30	31						
	16	17	18	19	20	21	22	23	24	25	26	27	28 School holidays	29 School holidays	30 School holidays				
ol ys	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	18	19	20	21	22	23	24	25	26	27	28	29	30						
ol ys	16 School holidays	17 School holidays	18 School holidays	19 School holidays	20 School holidays	21 School holidays	22 School holidays	23 School holidays	24 School holidays	25 Christmas Day	26 Boxing Day	27 School holidays	28 School holidays	29 School holidays	30 School holidays	31 New Year's Eve			
ol ys	13 School holidays	14 School holidays	15 School holidays	16 School holidays	17 School holidays	18 School holidays	19 School holidays	20 School holidays	21 School holidays	22 School holidays	23 School holidays	24 School holidays	25 School holidays	26 Australia Day	27 School holidays	28	29	30	31

Page Pro 99

Page Pro 99 is the remarkable page-making program for the TI-99/4A and Myarc Geneve 9640 computers that is unlike any other such program available. Unlike any page-making program for the 99/4A, or most for the Geneve, the page you see on the screen is exactly as it will appear when printed out.

Page Pro 99 is fast (written in assembly), easy-to-use (with extensive documentation and on-screen help), and powerful (a built-in TI-Writer editor and dozens of features unique to this program). It is compatible with standard TI-Artist fonts and pictures, as well as a growing library of *Page Pro 99* specific artwork (unlike other page programs *Page Pro* can use pictures as large as the page - all others are limited to small pictures). Page Pro will also work on any system from a TI-99/4A with a single disk drive to a Geneve with hard disks. It includes 3 thorough manuals with tutorials. It is very dependable and well-tested (as it should be after over 2 years of development). It is the best supported of any page-making program with dozens of packages of fonts, pictures, pre-made templates for all uses (labels, greeting cards, envelopes, business forms, etc.), borders and utilities. And finally, it produces beautiful pages.

Because it is "what-you-see-is-what-you-get", it is easy-to-use. Because it includes numerous examples, a good collection of starter fonts and pictures, and in-depth tutorials, you can use it straight out of the box. Because it is by Asgard Software, it has the support that it, and you deserve. Who can ask for more?

Page Pro Times

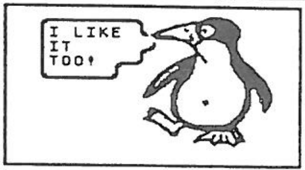
No.2 Winter '90

PAGE PRO 99 RELEASED!

ROCKVILLE, MD- Asgard Software recently released its long-awaited update to Page Pro 99 today to a thunderous standing ovation as members of the press viewed the many new features and capabilities of Version 1.6 of this popular program.

Hundreds of viewers clamoured to be among the first to receive the update to this package. After handing out updates, Asgard surprised participants by breaking out party favors and champagne for a celebration.

After the party, that lasted into the wee hours, and at its peak featured Asgard President Chris Bobbitt serenading his wife on his saxophone, accompanied by several members of the press using empty bottles as percussion instruments, the whole affair was broken up by local citizens and police outraged at not being invited. Crowds threw floppy disks at the celebrating throng. No arrests were reported but reportedly Mr. Bobbitt was offered a recording contract.



PROGRAM FEATURES

- 28 Pictures/Page
- Large font & small font on each page
- 2 Linestyles/page
- Print in single, double or "hi-res"
- Type up/down/left or right
- Import/Export TI-Writer text
- Save/load Page
- Use TI-Artist, other artwork
- Catalog disks
- Clip areas and save as pictures
- Much more!

ONLY $24.95
U.S. Add $0.75 S&H
Can. Add $1.25 S&H
Airmail Add $4.00 S&H

Asgard Software
Box 10306
Rockville,
MD 20849

- Paste up to 28 pictures of any size on a page at once
- Type in one large font and one small font per page
- Two different linestyles can be used at once
- Print out pages in single, double sharp "hi-res" mode
- Type in any direction - left, right, up or down
- Catalog a disk at any time while loading files
- Import/Export TI-Writer text - even columns of text
- Clip portions of the page and save as picture - reusable graphic elements!
- Includes utilities for converting fonts as well as creating two column text
- Use TI-Writer keys for editing text, place pictures where-ever the cursor is
- Useful for forms, letterheads, greeting cards, business cards, labels, signs, certificates, envelopes, etc.
- Requires 32K, Disk, and Epson or compatible printer

Asgard Software
P.O. Box 10306
Rockville, MD 20849
(703)255-3085

Only
$24.95

U.S. Add $0.75 S&H
Canadians Add $1.25
Europe/Asia/Australia
please add $4.00 S&H

Refer to page 4 for TIsHUG shop prices

Author	Title	Description	Page No.
McCormick,Mack			
USA	Assembler tutorial #1	Software hints	10.23
McGovern,Tony	Assembly squeezing, part 1	Software hints	7.22
	Assembly squeezing, part 2	Software hints	9.17
	Assembly squeezing, part 3	Software hints	10.06
	Assembly squeezing, part 4	Software hints	11.08
	Assembly squish	Software hints	7.21
	Extended BASIC tutorial	Software hints	1.09
	Extended BASIC tutorial	Software hints	2.09
	Extended BASIC tutorial	Software hints	3.07
	Living with spiders, part 1	Software hints	7.17
	Living with spiders, part 2	Software hints	8.15
	Living with spiders, part 3	Software hints	9.15
	PC-clone power supplies	Hardware hints	11.23
	Power supplies and transfomers	Hardware hints	8.17
	TI-Writer once again	Software hints	7.15
Moore,Fred			
CA,USA	Configure printer for DM1000	Software hints	11.17
Moseid,William			
USA	CorComp disk controller	General interest	4.27
Mudie,Ross	Assembly class	General interest	9.14
	Assembly class	General interest	10.09
	Assembly class	General interest	11.09
	Assembly tutorials	General interest	8.02
	Communicators	BBS information	4.13
	Communicators	BBS information	6.04
	Communicators	BBS information	7.09
	Disk file formats	Software hints	10.18
	Extended BASIC loaders	Software hints	8.09
	Extended BASIC trace to printer	Software hints	5.18
	Letter to editor	Monitors	6.02
	Linkit #15	Software hints	11.09
	New location for Sydney meetings	Club news	2.12
	Program information	BBS information	7.09
	Summer storms, computers, fax	General interest	11.09
	Uses for my computer	General interest	10.26
	Uses for your computer?	General interest	11.10
Murphy,Terrence	c99 TIPS slide show	Software review	11.15
USA	c99 TIPS to TI-Artist convert	Software review	11.15
	c99 TIPS viewer	Software review	11.15
Nielsen,R			
Qld	Viatel, what is it?	General interest	10.10
Ockenden,Eric	Monitor interface power supply	Hardware hints	6.07
Owen,John			
TX,USA	Configure printer for Fweb 4.10	Software hints	11.24
Paine,John	Overcoming alpha-lock	Hardware hints	4.16
Peterson,Jim	4-D graphics	Software hints	7.22
OH,USA	4-D graphics	Software hints	8.20
	Correction to tips #58	Software hints	5.05
	DataWriter version 1.2	Software review	5.04
	Games catalogue	Software review	9.11
	Programs to write programs #6	Software hints	6.16
	Putting it all together #6	Software review	5.18
	TI99/4A user groups	General interest	2.16
	Tips from the tigercub #38	Software hints	1.19
	Tips from the tigercub #39	Software hints	2.21
	Tips from the tigercub #40	Software hints	3.19
	Tips from the tigercub #41	Software hints	4.19
	Tips from the tigercub #59	Software hints	6.17
Pewterware			
USA	Periscope instructions	Game	1.11
Phillips,Terry	Agenda for 1990 AGM	Club news	1.02
	Minutes of 1989 AGM	Club news	1.03
	Minutes of 3rd AGM	Club news	10.03
	Minutes of special GM	Club news	10.04
	Secretary's notebook	Club news	1.04
	Secretary's notebook	Club news	2.03
	Secretary's notebook	Club news	3.02
	Secretary's notebook	Club news	4.03
	Secretary's notebook	Club news	5.03
	Secretary's notebook	Club news	6.04
	Secretary's notebook	Club news	7.04
	Secretary's notebook	Club news	8.04
	Secretary's notebook	Club news	9.03
	Secretary's notebook	Club news	10.05
	Secretary's notebook	Club news	11.03
	Secretary's report to 6/90	Club news	10.05
	TIsHUG software column	Club software	1.02
	TIsHUG software column	Club software	2.03
Presser,Joe			
USA	Getting a line on your program	Software hints	1.25
Pulley,Clint			
Ont,Canada	Bases in c99	Software hints	6.12
Raguse,Earl			
CA,USA	Beginning Forth #1	Software hints	11.17
Rebel,Eric-Paul	REF/DEF table	Software hints	2.23
Netherlands	Small list	Software hints	3.27
Reiss,Jim			
USA	Pix pro	Software review	5.07
Relyea,Bob	Cataloguing with multiplan	Software hints	7.19
	Editor's comment	General interest	3.01
	Editor's comment	General interest	4.01
	Editor's comment	General interest	5.01
	Editor's comment	General interest	6.01
	Editor's comment	General interest	7.01
	Editor's comment	General interest	8.01
	Editor's comment	General interest	9.01
	Editor's comment	General interest	10.01
	Editor's comment	General interest	11.01
	Editor's reply	General interest	8.03
	From the editor	General interest	2.02
	My use for computer	General interest	11.10
	Newsletter update	General interest	3.31
	Newsletter update	General interest	5.22
	Newsletter update	General interest	8.19
	TI-Base files to DV80	Software hints	6.12
	TI-Writer hints	Word processing	2.04
	Teachers use the TI99/4A	General interest	5.19
Roseen,Richard			
NJ,USA	Hardware corner	Hardware hints	11.26
Rosenberg,Howie			
USA	Brain, a review	Software review	1.10
Ruggeri,Alf	Greeting cards	Software hints	10.13

Author	Title	Description	Page No.
Saunders,Larry	Printer CTRL[U] commands	Word processing	4.21
	TI-Base tutorial	Software review	4.09
Scheidemantle,Paul			
OH,USA	Page Pro utilities	Software review	9.07
Schreiber,Rolf	Advertising really works!	General interest	11.24
	BOOT, bug fix	Software hints	7.06
	Cataloguing software library	Software hints	11.04
	Emulate file creation	Software hints	2.33
	Expanding your PEB	Hardware hints	9.20
	More PEEKs and POKEs	Software hints	2.07
	Ordering library software	Club news	5.04
	Software extra	Club software	2.03
	TIsHUG software column	Club software	3.03
	TIsHUG software column	Club software	4.06
	TIsHUG software column	Club software	5.04
	TIsHUG software column	Club software	6.06
	TIsHUG software column	Club software	7.05
	TIsHUG software column	Club software	8.05
	TIsHUG software column	Club software	9.05
	TIsHUG software column	Club software	10.07
	TIsHUG software column	Club software	11.05
Schubert,Peter	Multi-function card features	Hardware review	6.08
Scorpia	Cutthroats part 1	Adventure hints	7.10
USA	Cutthroats part 2	Adventure hints	9.06
	Cutthroats part 3	Adventure hints	10.26
	Sorcerer part 1	Adventure hints	2.15
	Sorcerer part 2	Adventure hints	3.10
	Sorcerer part 3	Adventure hints	4.31
	Sorcerer part 4	Adventure hints	5.19
	Spellbreaker part 1	Adventure hints	11.12
Shaw,Stephen	Bulletin board users	General interest	7.09
England	Disk reviews	Software review	7.14
	Disk reviews	Software review	10.17
	Education Modules	Software review	6.15
	Enhanced BASIC	Software hints	10.11
	G language	Software review	6.11
	Games modules	Software review	7.13
	Graphics for Myarc XB	Software hints	11.20
	High resolution graphics	Software hints	3.28
	Module reviews	General interest	4.28
	Ontplopper!	Software review	9.17
	Programs from TI*MES library	Software review	3.25
	Quick Run review	Software review	2.22
	Rambles	General interest	2.13
	Rambles	General Interest	5.15
	Rambles	General interest	6.15
	Rambles	Software hints	7.11
	Rambles	Software hints	11.11
	Reviews	General interest	3.23
	Reviews	General interest	4.17
	Super Extended BASIC	Software review	4.11
	Text on disk	Software hints	10.22
	TI Artist plus	Software review	3.14
	TI*mes disk library report	General interest	6.13
	UK users annual meeting	General interest	8.22
	Vector graphics and transforms	Software hints	11.21
Sheehan,Craig	Burning EPROMs	Software review	4.04
	EPROM RAMdisk	Hardware project	8.07
Slattery,Mike	BOOT, program loader	Software review	5.10
Smoley,Martin	TI-Base tutorial	Data base	1.13
OH,USA	TI-Base tutorial	Data base	2.11
Swedlow,Jim	Extended BASIC tips #1-4	Software hints	10.12
CA,USA	Extended BASIC tips #15	Software hints	8.13
	TI-Bits #1	Software hints	10.12
	TI-Bits #13 and #15	Software hints	3.18
	TI-Bits #16	Software hints	8.14
	TI-Bits #17	Software hints	8.14
	TIdBits #2	Software hints	11.16
	XB tips #14	Software hints	4.13
Takach,Ben	3.5 inch drives	Hardware hints	8.12
	Computers are innocent!	General interest	8.04
	Diagnosing simple HFDC fault	Hardware hints	6.10
	Future of TIsHUG	Club news	4.02
	Testing PC-Transfer	Software review	4.12
	TI hacker's lament	General interest	8.12
Trott,Geoff	Colour Monitor interface	Hardware project	1.05
	From the service bench	AT disk contr,RAMdisk	2.06
	From the service bench	Consoles	6.21
	From the service bench	RAMdisk	3.29
	How does it work?	General interest	2.34
	How does it work?	General interest	3.29
	How does it work?	General interest	4.29
	How does it work?	General interest	5.22
	Newsletter annual report	Club news	2.01
	Newsletter update	General interest	4.30
	Sheehan EPROM disk	Hardware hints	11.08
	They're off	General interest	1.01
	Treasurer's report	Club news	4.07
	Treasurer's report	Club news	5.03
	Treasurer's report	Club news	6.06
	Treasurer's report	Club news	7.08
	Treasurer's report	Club news	9.17
	Treasurer's report	Club news	10.21
	Treasurer's report	Club news	11.17
Warburton,Dick	Co-ordinators report	General news	1.02
	Co-ordinators report	General news	2.02
	Co-ordinators report	General news	3.02
	Co-ordinators report	General news	4.02
	Co-ordinators report	General news	5.02
	Co-ordinators report	General news	6.02
	Co-ordinators report	General news	7.02
	Co-ordinators report	General news	8.02
	Co-ordinators report	General news	9.02
	Co-ordinators report	General news	10.02
	Co-ordinators report	General news	11.02
Weasel,Inc.	Assembler titbit	Software hints	9.03
Wilkinson,Derek	Hard disk setting up	Software hints	7.08
Wolcott,Ron	TIPS documentation	Software review	11.13
Author unknown	Agenda of 4th AGM	Club news	10.04
	Author index 1989	General interest	1.15
USA	c99 function library	Software hints	3.27
USA	Disk peripheral, part 1	Specifications	3.21
USA	Disk peripheral, part 2	Specifications	4.25
USA	Forth to you too!	Software hints	1.24
USA	Forth to you too!	Software hints	2.20
USA	Forth to you too!	Software hints	3.24

GPL, an introduction

Extracted from Texas Instruments manual, USA

1.0 GRAPHICS PROGRAMMING LANGUAGE

The System software resident in the product consists of a monitor and a GPL (Graphics Programming Language) processor. It is the function of the monitor to insure that every time the system is turned on, a new cartridge is inserted, or an existing program terminates, that all memory and peripheral devices are initialized. The GPL processor is an interpreter optimized to execute GPL programs directly out of GROM. The GPL processor software is coded in TMS 9900 assembly Language. NOTE: See Appendix N.

1.1 OVERVIEW

GPL is a programming language specially developed by Texas Instruments to provide the best possible tradeoff of code compaction, execution speed, and ease of program development for the target computer system. The GPL instruction set facilitates development of programs which make use of the unique features of the system chip set. It is byte oriented, and instructions typically have one or two operands. The addressing scheme is such that most instructions can access either standard microprocessor RAM, GROM, or the video scratchpad RAM address space easily.

Most instruction operands can be either single or double byte values. The addressing modes are: immediate, direct, indirect, indexed, indexed indirect (with pre-indexing), and 'top of stack'. Source operands and destination addresses can be in the CPU, video RAM, or in GROM. Support for two stacks is available; a data stack and a subroutine return address stack (allowing arbitrary nesting of subroutines).

1.2 GPL INSTRUCTION SYNOPSIS

GPL has the following types of instructions:

*DATA TRANSFER	-single or double byte transfers -block to block transfers -formatted block transfers
*ARITHMETIC	-add, subtract, multiply, divide, -negate, absolute value
*LOGICAL	-and, or, exclusive or, shifting
*CONDITION TESTS	-arithmetic and logical tests
*BRANCHING	-unconditional and conditional
*BIT MANIPULATION	-set, reset, and test
*SUBROUTINING	-call, return, parameter fetching
*STACK OPERATIONS	-push and pop
*MISCELLANEOUS	-random number generation, keyboard scan, -coincidence detection, pattern movement, -sound control, -TMS 9900 subroutine linking, I/O

1.3 GPL TIMING

The GPL interpreter contains an interrupt driven service routine which is tied to the video scan. Video symbols may be moved about the screen automatically; also sounds may be generated from a sequence table.

These are of the "set it and forget it" type of instructions which free up the control program to do concurrent decision and computational operations. The interrupt also controls a software real time clock.

Each system will have a clock byte reserved in the console ROM at location >000C to indicate the clock rate for that system. Peripherals may read this byte to adjust their timing interface to the CPU's clock combinations in different consoles. The high nybble contains the integer frequency in megahertz and the low nybble, the fractional frequency.

1.4 GPL ASSEMBLER

The TI assembler for GPL (GPLASM) is written in a mixture of FORTRAN and assembly language and is currently available for installation on 990/10 DS minicomputers. The assembler provides standard features such as creation of a list file, cross reference tables, and error flagging. A set of macros is included to help structure GPL programs; these include statements such as: REPEAT ... UNITL and IF ... THEN ... ELSE. The output of the assembler is a 990 object module.

1.5 SOFTWARE MONITOR RECONFIGURATION

The monitor code is executed whenever a system restart is required. The system parameters and control values are initialized to default values. A default character set is loaded into the video pattern generator, making it immediately available to GPL programs. This pattern set consists of 64 ASCII characters, including the upper-case alphabet, digits, arithmetic symbols, and punctuation symbols.

The monitor is also responsible for determining the existing system configuration. The power-up monitor must poll add-on I/O peripherals and the "SOLID STATE SOFTWARE CARTRIDGE" to determine which program to execute.

The Home Computer system has been designed to be flexible and expandable. Each plug-in ROM or GROM may contain power-up procedures. These power-up procedures will all be executed allowing for expansion of the power-up routines. A power-up routine may also be replaced by another.

1.6 FOREIGN LANGUAGE SCREENS

GPL code has been included in GROM 0 to allow a plug-in GROM to "translate" the main screen, the menu screen, and the cassette DSR messages to alternate languages. The main screen and the menu screen are "translated" after the screen has been formatted in English but while the screen is turned off (only the background colour is visible on the screen). At this time, the plug-in GROM is checked for a negative version number (byte 1 of the GROM). When a negative version is encountered, a GROM routine is called at >6010 for the main screen or >6013 for the menu screen. These locations should contain unconditional branches to the routines in the plug-in GROM that will rewrite the screen in the desired language. These routines may use all of the usual CPU RAM locations (>0 through >6F) and the full facilities of the monitor and interpreter. The routines should end with a RTN instruction. o

continued from page 12

Author	Title	Description	Page No.
Author unknown	From the bulletin board	Mail to all	1.24
	From the bulletin board	Mail to all	2.30
	From the bulletin board	Mail to all	3.22
	From the bulletin board	Mail to all	4.23
	From the bulletin board	Mail to all	5.09
	From the bulletin board	Mail to all	11.10
USA	Handy hints	Software hints	11.18
	Modified formatter commands	Word processing	2.24
	Notice of General Meeting	Club news	6.03
	Program to type in	Grue stew	3.15
USA	Program to type in	Horse sprite	9.18
USA	Real programmers, part 1	General interest	8.21
USA	Real programmers, part 2	General interest	9.21
Author unknown	Regional group reports	General interest	1.31
	Regional group reports	General interest	2.35
	Regional group reports	General interest	3.31
	Regional group reports	General interest	4.31
	Regional group reports	General interest	5.23
	Regional group reports	General interest	6.23
	Regional group reports	General interest	7.23
	Regional group reports	General interest	8.23
	Regional group reports	General interest	9.23
	Regional group reports	General interest	10.27
	Regional group reports	General interest	11.27
	Subject index 1989	General interest	1.14
USA	Symphony	General interest	6.05
	Tetris, a game	Software review	3.30

o

Appreciate your Programmers

Spell It!, a review

by Geoff Trott

Spell It! is a new program for all who use word processors for writing English. It provides the ability to check the spelling of any document produced by a word processor. As such it is a very useful program and I would recommend that everyone buy a copy and use it always. Even those of us who can spell, occasionally make mistakes, both typographical or finger mistakes and spelling errors. Most people I know do not like reading their own masterpieces and even if they do, it is hard to pick up your own errors. Running your text through a spelling checker gives an independent check which picks up most of the silly mistakes. For example, in the documentation for the latest version of Funnelweb (4.31) there are a few places where q is used in place of a (they are very close to each other on the keyboard). Using one of the character sets, it is quite difficult to spot the difference on the screen as the q does look like an a on the screen, but the spelling checker quickly finds these problems.

While editor of the TND, I became used to using a spelling checker and quickly recognised some general desirable features of spelling checker programs. They should run quickly as you do not want to wait too long while words are checked. They should be able to handle any document produced with TI-Writer and its variants. They should not have abbreviations in the dictionary in case you do not want to use abbreviations and they should not have any spelling errors in the dictionary. They should not have special words like the names of the states of the USA either. In fact I am convinced that the main dictionaries should not be exhaustive but should just cover the main words with any jargon or less usual words in a user dictionary which can be used if wanted. There are always words that will not be in the dictionary that you will want to use, for example names. However it is annoying if some normal words are not in the dictionary so that the dictionaries should be as complete in that respect as possible. Of course, the more words there are in the dictionary, the longer it takes to check a document. It is also essential to be able to view the disputed word in context to make sure of what is required and it is nice to have the program make a suggestion as to the word required.

Now let me tell you how I see Check It! measuring up to these requirements. Let me make it plain at the outset that I think that this program is excellent and as I have already said one that everyone should buy. However, in doing a review, I feel I must tell you all the bad things and a lot of the good things are more or less understood. For example, the program works well, has good documentation, runs quite quickly and can be configured to run on a range of systems, making use of RAMdisks and hard disks to speed up the processing where possible. However, there are some areas which are not as good as they could be and unfortunately they need a lot more words to explain than all the good things combined. This is the problem with doing an in depth review and I hope you will understand what I am trying to say.

Now to the actual program. The version I used was the DSDD version which I put on an emulated DSK1 on my hard disk. The files which I used to test the program were in a sub-directory on the hard disk. I first ran the CONFIGURE program. This allows for 40 or 80 columns, a path name for the dictionaries, a path name for the text files and selection of a character set. It does not seem to have a way of setting colours only using black on grey. The program can be set up to be called from the menus used by your favourite word processor. Once it has been configured, the program can be run.

The program asks for the file to be checked with the path specified already there. When the file name has been entered, the file is read in and a list of unique words is set up. The whole document is not stored in memory so a file which is too big to handle by TI-Writer can be checked for spelling. The program tells you how many unique words it found and the total number of words read which is very useful. Then it goes off to check all the words against its dictionaries. While this is going on its progress is reported on the screen. When it finishes the number of unknown words is reported and the program waits for a key press.

Now the list of words not known are presented one at a time for action. At each word you can view the word in context, correct the word, add the word to the dictionary, see what words are in the dictionary like the unknown word, move on to the next word or move to the previous word. If you add a word to the dictionary then that word is no longer available in the list of unknown words. You cannot change a word which you have already changed. This is where I would like to see a few changes to the program. Firstly, the words are presented as all upper case. If you view them you can see the correct case, but only the first occurrence of the word. In fact the view function has a few problems. It sometimes does not find the word because of other special characters around the word. For example, a word surrounded by an ampersand at the start and a defined space (caret) at the end was found by the first part of the program, but not by the view, nor by the correction part of the program. The other problem with the view is that it shows the disk sector which contains the word and so the word sometimes may be the first word in the sector or the last word in the sector. The words are examined in the order of their appearance in the document which makes it difficult to know how far from the end you are. Corrections are typed in in upper case and then take the case of the word being replaced which does not allow the case of the words to be changed. For example, if you wanted to replace XB by Extended BASIC, it would not be able to do that. You will end up with EXtended basic. One good feature is that the formatter commands that start with a "." or ";" are ignored but words with either { or } at the start or end are not ignored.

The ability to check the dictionaries for help in the correct spelling is also very useful but sometimes does not give much information. Words with only one syllable give a number of possibilities while words with many syllables usually give none. Also, if the first letter is not correct then the wrong dictionary is searched.

Once the list of words is dealt with, the program makes the changes by reading in the original file, making any changes and saving the result to another file with an S appended to the file name. There is another problem here if you make a change which increases the number of characters, you may lose characters from the end of that line if the new line ends up longer than 80 characters. (Changing XB to Extended BASIC for example.) The program then returns to ask for another file to be checked, which I think is great.

There are some minor problems such as American spelling and words not in the dictionaries, but at least the dictionaries are in a form that they can easily be modified. Dictionary file DC has an error (COMPONENTL) as does QUIK (ESPRESS and TECHINCAL).

Clearly there are a few problems which could be easily fixed in the program and some others in the contents of the dictionaries which are able to be fixed by the user. Apart from these problems the program runs well. It is faster than the other spelling checker I am using and provides an easier path to tidy up the dictionaries. It is far more user friendly than the other spelling checker with less typing required to run it and more useful information returned. I like the word counts, the ability to set a character set, the default path and the fact that when it finishes one file it asks for the next one. It would be nice to have some control over the colours and for the bugs to be removed but as I have said already I would recommend everyone who is word processing to buy and use a copy.

Tips from the Tigercub
#60

by Jim Peterson, Tigercub Software, USA

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in Supplement #8.

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00. If I run out of printed documentation, it will be supplied on disk.

My TI-PD library now consists of 400 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #2 with Supplement #8, listing all titles and authors, is currently available for $1 which is deductible from the first purchase.

Here are a couple of improvements to the CHARFIX subprogram published in Tips #58.

```
29000 SUB CHARFIX(HX$()):: DISPLAY AT(12,1)ERASE ALL
BEEP:"Transliterate Punctuation?" :: ACCEPT AT(12,28)
SIZE(1)VALIDATE("YN"):Q$ :: IF Q$= "N" THEN 29004
29007 CALL CHARVIEW(HX$())
29009 SUB CHARVIEW(HX$())
```

And call the routine by CALL CHARFIX(HX$()). These changes will avoid unwanted transliteration, and will make it possible to use CHARFIX for ASCII 24-31 and 144-159, if BXB has been merged in, as described in Tips #55.

The Spring 1990 issue of the TI*MES newsletter from England contained an interesting challenge - write a program in any language to find the lowest power of 7 which contains six sevens in succession, i.e. "777777".

The computer cannot solve this by any normal means, because it soon goes into scientific notation in which large numbers are rounded off into long strings of zeros. So, I taught it to multiply the old-fashioned way -

```
100 A$=STR$(7):: Y=1
110 Y=Y+1 :: FOR J=LEN(A$)TO 1 STEP -1 :: E=(VAL(SEG$
    (A$,J,1))*7+X)/10
120 X=INT(E):: F=(E-X)*10 :: X$=STR$(F)&X$ :: NEXT J
130 IF X>0 THEN X$=STR$(X)&X$
140 IF POS(X$,"777777",1)<>0 THEN S60
150 A$=X$ :: X$="" :: X=0 :: GOTO 110
160 PRINT "7^";STR$(Y);"=";X$
170 PRINT #2:"7^";STR$(Y);"=";X$
```

The answer? 7^175=7801120791220815810240464127911180777777188182006932636111839698571603885844026671779915606471699893312656644407347632248554716494939953912586437943

My TI-99/4A computed that in 24 minutes. Would someone like to try it on the 9640?

Anyway, I thought I would use the same method to solve precise multiplication of numbers too large to be computed directly. This routine will multiply two numbers of up to 28 digits each, and will handle decimals and negative numbers. For even larger numbers, change the ACCEPTs to INPUTs and if necessary change the DIM. The only limitation seems to be that the result cannot contain more than 256 digits and even that could be programmed around.

```
100 DIM C$(100)
110 DISPLAY AT(12,1)ERASE ALL:"FIRST NUMBER?" :: ACCEPT
AT(14,1)VALIDATE(NUMERIC)BEEP:A$
120 IF SEG$(A$,1,1)="-" THEN A$=SEG$(A$,2,255):: M=1
130 A=LEN(A$):: D1=POS(A$,".",1):: IF D1>0 THEN A$=SEG$(
A$,1,D1-1)&SEG$(A$,D1+1,255) :: D1=A-D1
140 DISPLAY AT(16,1)ERASE ALL:"SECOND NUMBER?" :: ACCEPT
AT(18,1)VALIDATE(NUMERIC)BEEP:B$
150 IF SEG$(B$,1,1)="-" THEN B$=SEG$(B$,2,255):: M=M+1
160 Y=LEN(B$):: D2=POS(B$,".",1):: IF D2<>0 THEN B$=SEG$
(B$,1,D2-1)&SEG$(B$,D2+1,255):: D2=Y-D2 :: D1=D1+D2 ::
Y =Y-1
170 FOR J=Y TO 1 STEP -1 :: W=W+1 :: B=VAL(SEG$(B$,J,1))
:: FOR K=LEN(A$)TO 1 STEP -1 :: A=VAL(SEG$(A$,K,1))
180 D=(A*B+X)/10
190 E=INT(D):: F=(D-E)*10 :: C$(J)=STR$(F)&C$(J):: X=E
:: NEXT K
200 IF X>0 THEN C$(J)=STR$(X)&C$(J)
210 C$(J)=C$(J)&RPT$("0",W-1)
220 X=0 :: NEXT J
230 L=LEN(C$(1)):: FOR J=1 TO Y :: L2=LEN(C$(J)):: IF
L2 < L THEN C$(J)=RPT$("0",L-L2)&C$(J)
240 NEXT J
250 FOR J=LEN(C$(1))TO 1 STEP -1 :: FOR K=1 TO Y :: G=G+
VAL(SEG$(C$(K),J,1)):: NEXT K
260 G=(G+H)/10 :: L=INT(G):: G=(G-L)*10 :: D$=STR$(G)&D$
 :: H=L :: G=0 :: NEXT J
270 IF H>0 THEN D$=STR$(H)&D$
280 IF D1>0 THEN D$=SEG$(D$,1,LEN(D$)-D1)&"."&SEG$(D$,LE
N(D$)-D1+1,255)
290 IF M=1 THEN D$="-"&D$
300 PRINT D$
```

And this one will add up an almost unlimited number of integers of almost any length - I have not figured out how to get it to line up decimals.

```
100 CALL CLEAR :: DIM C$(100)
110 DISPLAY AT(12,1):"Input from D":" (D)isk or":" (K)
eyboard?" :: ACCEPT AT(12,12 )VALIDATE("DK")SIZE(-1):Q$
:: IF Q$="K" THEN 140
120 DISPLAY AT(12,1)ERASE ALL:"Filename? DSK" :: ACCEPT
AT(12,14):F$ :: OPEN #1:"DSK"&F$,INPUT
130 X=X+1 :: LINPUT #1:C$(X) :: M=MAX(M,LEN(C$(X)):: IF
EOF(1)<>1 THEN 130 ELSE CLOSE #1 :: GOTO 160
140 DISPLAY AT(12,1):"Press ENTER when finished":"":""
150 X=X+1 :: INPUT C$(X):: M=MAX(M,LEN(C$(X))):: IF C$(X
)<>"" THEN 150 ELSE X=X-1
160 FOR J=1 TO X :: IF LEN(C$(J))<M THEN C$(J)=RPT$("0",
M-LEN(C$(J)))&C$(J)
170 NEXT J :: FOR J=M TO 1 STEP -1 :: FOR K=1 TO X :: G=
G+VAL(SEG$(C$(K),J,1)):: NEXT K
180 G=(G+H)/10 :: L=INT(G)::G=(G-L)*10 :: D$=STR$(G)&D$
 :: H=L :: G=0 :: NEXT J
190 IF H>0 THEN D$=STR$(H)&D$
200 PRINT D$
```

It is easy to invert characters on the screen simply by making the foreground "on" pixels a lighter colour than the background "off" pixels - but when you make a screen dump, you will find that the "on" pixels will print and the "off" pixels will not.

Key this in, SAVE it by SAVE DSK1.INVERSE,MERGE and then merge it into any program by MERGE DSK1.INVERSE, call it at any point by CALL INVERSE(A,B), (A and B are the first and last ASCII to be inverted), and you will have all "on" pixels turned off and vice versa.

```
31111 SUB INVERSE(A,B):: FOR CH=A TO B :: CALL CHARPAT(C
H,CH$)
31112 FOR J=1 TO 16 :: CH2$=CH2$&SEG$("FEDCBA9876543210"
,POS("0123456789ABCDEF",SEG$(CH$,J,1),1),1):: NEXT J ::
CALL CHAR(CH,CH2$):: CH2$="" :: NEXT CH
31113 SUBEND
```

Here is a truly remarkable discovery by Bill Hudson of the Central Ohio Ninety Niners. This 2-line program will allow you to RUN a variable name such as - A$="DSK1.PROGRAM"

You can write lines before these, after these, and even RES the program. You can also use MOVE from GK UTILITY. You can do anything to the program you want as long as you do not change the content of line 1000. The line number does not even have to be 1000 BUT IT MUST BE THE FIRST LINE THAT YOU KEY IN!! You can merge a program into this but cannot merge this into a program. Line 900 can also be a different line number but program execution must go to that line first.

```
900 FOR Z=1 TO LEN(A$):: CALL LOAD(-41+Z,ASC(SEG$(A$,Z,1
)),0):: NEXT Z :: CALL LOAD(-41,LEN(A$)):: CALL LOAD(-44
,4+LEN(A$))
1000 RUN "DSKx.1234567890"
```

It has been a long time since we had a screen display to watch just for the fun of it, so here is a tinygram -

```
100 CALL CLEAR :: FOR SET=1 TO 14 :: CALL COLOR(SET,SET+
1,SET+2):: NEXT SET :: CALL SCREEN(2):: CALL VCHAR(1,1,3
1,768)
110 FOR CH=32 TO 136 STEP 8 :: CALL CHAR(CH,"FF000000000
000FF"):: NEXT CH
120 X=INT(RND*6+1)*2-1 :: Y=INT(14*RND+1)*8+32 :: FOR R=
12-X TO 12-INT(RND*X):: CALL HCHAR(R,5,Y,R)
130 CALL HCHAR(25-R,5,Y,R)
140 CALL HCHAR(R,28-R,Y,R)
150 CALL HCHAR(25-R,28-R,Y,R)
160 ON INT(2*RND+1)GOTO 170,190
170 CALL HCHAR(R,4+R,Y+8,25-R*2)
180 CALL HCHAR(25-R,4+R,Y+8,25-R*2)
190 NEXT R :: GOTO 120
```

This is a challenging and educational math puzzler which I think is unlike anything you have seen. I had it in my Tigercub catalog for 7 years and sold just 18 copies. If you do not want to key it in, it is now one of the programs on TI-PD disk No. 1300.1.

```
100 GOTO 140
110 J,K,ST,LV,I,R(),T,X,ADA$,X$,B,B$,C,C$,D,D$,AY,BY,B@$
,BY$,CY,CY$,C@$,Q,Y(),Y@,X@(),FLAG,R$,RL,Z,YY,D@(),Q$
120 CALL CLEAR :: CALL CHAR :: CALL COLOR :: CALL VCHAR
:: CALL SCREEN :: CALL KEY :: CALL SOUND
130 !@P-
140 CALL CLEAR :: FOR J=1 TO 12 :: CALL COLOR(J,5,16)::
NEXT J
150 CALL VCHAR(1,3,32,672):: DISPLAY AT(5,1):" @$%#*#+#
RITHMATIK #+#%$@ "
160 DISPLAY AT(10,1):" Select difficulty level -": :"
type 1 or 2"
170 CALL KEY(0,K,ST):: IF ST<1 THEN 170
180 IF (K<49)+(K>50)THEN 170
190 LV=K-48
200 CALL VCHAR(1,3,32,672)::FOR I=1 TO 4 :: RANDOMIZE
210 R(I)=INT(RND*10) :: IF R(I)=0 THEN 210
220 FOR T=1 TO I-1 :: IF R(I)=R(T)THEN 210
230 NEXT T
240 NEXT I :: X=R(1)*1000+R(2)*100+R(3)*10+R(4)
250 A=INT(4*RND)+1
260 ON A GOSUB 330,340,350,360 :: A$=X$
270 B=INT(4*RND)+1 :: IF B=A THEN 270
280 IF (LV=1)*(LEN(STR$(R(B)/R(A)-INT(R(B)/R(A))))>2)
THEN 250
290 ON B GOSUB 330,340,350,360 :: B$=X$
300 C=INT(4*RND)+1 :: IF C=A THEN 300
310 IF C=B THEN 300
320 ON C GOSUB 330,340,350,360 :: C$=X$ :: D=10-A-B-C ::
ON D GOSUB 330,340,350,360 :: D$=X$ :: GOTO 370
330 X$=" 1st " :: RETURN
340 X$=" 2nd " :: RETURN
350 X$=" 3rd " :: RETURN
360 X$=" 4th " :: RETURN
370 AY=R(B)/R(A):: BY=ABS(R(C)-R(B)^2):: IF BY=0 THEN 38
0 ELSE 390
380 B@$="" :: BY$=" equal to " :: GOTO 400
390 B@$=STR$(BY):: BY$=" more or less than"
400 CY=ABS(R(D)-R(C)-R(B)-R(A)):: IF CY=0 THEN 410 ELSE
420
410 CY$=" equal to" :: C@$=" " :: GOTO 430
420 CY$=" more or less than" :: C@$=STR$(CY)
430 DISPLAY AT(2,1):" I have a 4-digit number ":" with
no two digits the":" same." :: DISPLAY AT(6,1):" The";
B$;"digit is";AY;" times the";A$;"digit."
440 DISPLAY AT(9,1):" The";C$;"digit is ";B@$;BY$;" the
square of the";B$;" digit." :: DISPLAY AT(14,1):" The";D
$;"digit is ";C@$;" ";CY$;"the sum of the other digits"
450 DISPLAY AT(18,1):" What is the number?" :: ACCEPT AT
(20,2)VALIDATE(DIGIT)SIZE(4) BEEP:Q :: IF Q=X THEN 530
460 Y(1)=INT(Q/1000):: Y(2)=INT((Q-1000*Y(1))/100):: Y(3
)=INT((Q/100-INT(Q/100))*10) :: Y(4)=(Q/10-INT(Q/10))*10
:: IF Y(B)<>INT(Y(A)*AY)THEN 570
470 IF BY<>0 THEN 490
480 IF Y(C)<>Y(B)^2 THEN 570 ELSE 500
490 IF (Y(C)<>Y(B)^2+BY)*(Y(C)<>Y(B)^2-BY)THEN 570
500 IF CY<>0 THEN 520
510 IF Y(D)<>Y(A)+Y(B)+Y(C)THEN 570 ELSE 530
520 IF (Y(D)<>Y(A)+Y(B)+Y(C)+CY)*(Y(D)<>Y(A)+Y(B)+Y(C)-C
Y)THEN 570
530 DISPLAY AT(22,1):" Correct!": : :: FOR J=1 TO 2 :: C
ALL SOUND(100,392,5):: CALL SOUND(100,440,5):: CALL
SOUND(100,494,5):: CALL SOUND(10 0,523,5)
540 NEXT J :: CALL SOUND(1000,523,5,392,5,330,5)
550 DISPLAY AT(24,1):" Hit any key"
560 CALL KEY(0,K,ST):: IF ST<1 THEN 560 ELSE 200
570 DISPLAY AT(22,1):" Wrong." :: CALL SOUND(900,30000,3
0,30000,30,400,30,-4,0):: DISPLAY AT(23,1):" Type A to
try again or Z":" to see the number"
580 CALL KEY(0,K,ST):: IF ST<1 THEN 580
590 IF K=65 THEN 450
600 IF K=90 THEN 610 ELSE 580
610 DISPLAY AT(22,1):" The number was";X:" " :: GOTO 550
:: END
```

Nearly out of memory and all out of ideas. More next time, maybe.

Jim Peterson
Tigercub

o

continued from page 6

Particularly impressive are the facilities which enable the user to search for a nominated string, ie. a word or phrase, and to replace strings with different ones at all or selected occurrences. Lines of text can be copied and can be easily moved around to different positions on the page.

Text can be saved to cassette or disk. Using the SaveFile -SF- option will store your outpourings in the usual Dis/Var (i.e. text) mode and by specifying start and finishing line numbers it will save only that part of the file. An interesting feature is that files can also be saved using SaveMem -SM- or SavefiX -SX- options. SM saves in memory image, ie program, mode thereby reducing the space needed, whilst SX stores in Dis/Fix format. Similarly files (or parts of files) can be loaded using LF, LM, or LX.

If you do have the facilities to save to disk another useful capability would be that of Show Directory (SD). This displays all files held on the specified disk, giving details of the stored format, sectors used and size of each file.

Finally, of course, you can print your file through the module interface or through to any other printer you may have connected to your computer. Printer control characters can be entered (escape sequences) to give access to your printer's special commands, eg. underlining or changing fonts.

It should also be noted that for most of the functions obtained by going through the command line, there is also the option of, what is termed in the perfectly adequate accompanying handbook, the Expert Mode. For example, instead of FCTN 9 & PF to get to PrintFile, this can be entered merely by keying CTRL P.

So, in summary, if you have been looking for a way to get a lot more out of your dusty, old TI by expanding into the word-processing field, but have shied away from a fully expanded system, this comprehensive module may be just the thing for you.

[Ashley now has a PEB and TI Writer. In order to load the Wordwriter+ DV80 files into TI Writer it is necessary to use the short conversion program in TI*MES #21, or to use VIEW to discover the last line number and then LOAD all lines EXCEPT that one - both these tricks divert TI Writer from the non-compatible Wordwriter+ control codes in the final record.] o

Beginning Forth - part 2

by Earl Raguse, UGOC, CA USA

I will endeavour to set forth (no pun intended!) the basics of the Forth language to the limit of my knowledge. From the organisational point of view see my learning plan outline below. I will assume only that you know how to operate your computer and that you have all necessary equipment connected and turned on.

WHAT IS THE NECESSARY EQUIPMENT?

Well, as a minimum you must have a console with monitor, 32k memory expansion, disk controller and at least one disk drive (I will only briefly touch on single drive operation, its a drag). You also must have a way of loading Forth. That can be done using Editor/Assembler or FunlWeb(Writer). There is a Forth disk available with a machine language loader included on the disk for access by XBASIC. If it is not now in the library, I will put it there. You can learn Forth without having an RS232 interface and a printer, but it sure is not much fun.

You must also have a TI Forth Manual (TIFM). I do not intend to parrot that back to you in its entirety. There is much it does not say, but what it does say is reasonably clear. Where I think it warrants, I will explain it as I see it. If you do not have one check our library, or talk to Chick De Marti.

LEARNING PLAN

1. Getting started, some terminology, the screen concept, fixing TI errors, making BSAVED Working Disks and Data Disks. Elementary words, printing text to the screen and printer.

2. Fixing the Editor to have an auto repeat cursor. You cannot appreciate this unless you have done without it.

3. Stack manipulations and doing integer arithmetic and logical operatons.

4. Learn how to define new words and use loops, both definite and indefinite.

5. Rewriting some Forth words to suit yourself and making some new ones. Better known as making life easy for yourself. Know where not to change Forth.

6. Floating Point Arithmetic (FPA) and how to live without it, unless you are desperate and not in a hurry.

7. Trigonometric functions etc. without FPA.

8. Graphics capability of Forth and some simple and not so simple examples.

9. Generating sound in Forth. There are no resident sound control words, but no matter, we will make our own, and some music too.

10. String handling words that we will define. Forth has very few resident ones, but we fix that.

11. Files and disk access, writing screens under program control. Elementary sorting with an animated demo.

12. Subscripted variables, multi-dimensional arrays and matrices. How to use compiler extension ability.

13. Making Forth talk, anything the 99/4A can do Forth can do.

14. Other stuff like Reverse Video highlighting, special requests etc.

WARNING-FEEDBACK REQUIRED

The above plan can and will be changed on request, if its reasonable and I know how. I will also probably change it to account for things I have forgotten, or new things I learn. I am always doing that.

Because of the time lag between writing of these articles and printing The TOPICS, three months of these articles will be prepared. If the author or the TOPICS editor does not receive an indication of interest in Forth, the author will know that he is beating a dead horse. If there is in fact interest, the articles will continue as long as you want them. I have been writing about Fourth for two years now and do not intend quit on you.

GETTING STARTED IN FORTH

In last month's article, I promised to get you started in Forth. This is because I believe that there are TIers out there who would like to try Forth but who are a little unsure how to proceed. They have either never tried it or have loaded it once when it first came out and have long since forgotten how to go about it. If you fall into either group, read on, this series of articles is for YOU.

AWAY WE GO!

These instructions assume you still have your original Forth System Disk in the same condition as you received it from the User's Group sometime back in the early 80's. For those of you who have gotten beyond that point, have patience, this will not hurt much and we will catch up to you quite soon.

To digress a bit, Forth stores programs and data on a disk in chunks of 1024 bytes (a nice round 1K, which is exactly 4 Sectors). These chunks are called "screens" and arranged as files of 16 lines of 64 bytes. Ninety (90) screens fit on a single sided single density (SSSD) disk. The original Forth System Disk comes with almost all of the 90 screens filled, some with optional utilities. Only those actually used need be loaded. The Forth kernal resides on Screens #8 through #19. Error message screens #4 and #5 are mandatory on any Forth disk in drive 1. Screens #0, #1 and #3 are also required for a bootable disk, otherwise a Forth disk may have all the rest of the screens blank for writing your programs.

If you elect to use the Data Disk concept to store your programs and data, then only Screens #4 #5 are required on the disk, and then only if it is in drive 1. I like to make all my disks bootable (ie a System Disk). Some may think of this as a waste of good disk space, but I call it convenience.

There is nothing magic about screen numbers, it is just an ID number. I will make example screens using numbers convenient for me. You may move them anywhere you like, except Screen #3 is special do not change that number. However, when a word references a screen number you must change it if you change screen numbers. Think of it as a BASIC subroutine, they can be on any line number so long as the calling statement has the right number.

TERMINOLOGY

I strongly recommend that you try to read the TI Forth Manual (TIFM), at least through to Chapter 5, before starting to follow what I do.

In the TIFM they make a point of trying to distinguish between a Forth "SCREEN" and the monitor "screen" by the use of capitals, but they do not stick with it. I will abandon this and capitalize only the first letter in "Screen" only if referring to a particular screen. The monitor screen will always be just CRT.

I think it unfortunate that the author of Forth chose to call Forth commands "words" because that can be a source of confusion. Considering the way they operate, I think it would be more appropriate to call them "verbs", but I probably will not. I am a conformist I guess.

From here on I will refer to the disk with the original Forth or any bootable copy of it, as the System Disk and an unbootable disk you use to store new programs or applications on, as the Data Disk. If you plan to use your Data Disk in Drive#1, Forth requires that you copy the error message Screens #4 and #5 using SCOPY.

Read again the TIFM Chapter 3, pages 1 and 2, especially if you have only one disk drive. On page 1, up 4 and 5 lines from the bottom, change "FORTH disk" to "Data disk". The procedure on page 2 makes more sense if you make the following changes:

1. In the START line, change "original diskette" to "System Disk".

2. In the LOOP section, change "backup diskette" to "Data Disk".

3. After the second FLUSH, change LOOP to START.

In Chapter 3, page 5, fix/add the following:

INSERT - Inserts a space where the cursor is.

ERASE - Deletes a complete line, but saves it on the pad for use by REDO.

REDO - REDO may be used to move lines to another screen.

So much for digression, let us get on with it. To begin with, there is the matter of correcting an error on your original Forth System Disk Screen #72, and changing it, if necessary , to suit your printer. Also we will correct some errors which have been found on Screens #53, 54, 55, 58, and 59. Then, you should make a BSAVE'd version of Forth, which is sometimes called "Fast Loading Forth".

FIRST, make a backup of your original Forth System Disk with Disk Manager, DM 1000, TurboCopy, or whatever. Beware, I have found that at times Forth disks copied with DM will not boot. Be sure to test them.

SECOND, put away your original Forth System Disk and use your backup to boot up Forth. To do this you need one of the following: Editor/Assembler module, FunlWeb(Writer), or possibly XBASIC. Recall that I told you about the XB version of Forth. I always use the E/A module. If you use FW just follow the menu prompts. For XB just sit back and let it load. What you will see will be a little different, but I think you will figure it out. Your system must also include the 32K memory expansion, at least one disk drive. A printer is nice but not actually essential.

(1) With E/A in the cartridge slot, power up the P-Box, and all other accessories you have, then the computer. Install your NEW unprotected Forth System Disk in Drive 1
(2) Select 2. EDITOR/ASSEMBLER
(3) Select 3. LOAD AND RUN
(4) Respond to file query with DSK1.FORTH
(5) You should get a blank screen, then BOOTING....
(6) In a few seconds, you should see the Forth Menu, which looks a little like this.

```
-SYNONYMS -EDITOR    -COPY
-DUMP     -TRACE     -FLOAT
-TEXT     -GRAPH     -MULTI
-GRAPH    -FILE      -PRINT
-CODE     -ASSEMBLER -64SUPPORT
-BSAVE    -CRU
```

(7) Enter -EDITOR <ENTER> (We will talk about -64SUPPORT later)
(8) Now enter 72 EDIT ~ENTER> (This is the last time I will mention the need to use <ENTER> after an entry)

EDITING

You should now have Screen #72 on the CRT. Notice that on line 5 you have "PAB_ADDR". Move the cursor by pressing the E S D X arrow keys while holding down the FCTN key, until you get to the "_". Replace the "_" with "-" by typing over it to get "PAB-ADDR". The error is fixed. Now you need to save the fix. To do that press FCTN 9 (Back) to exit the EDITOR, the cursor should move to the bottom of your screen, then type FLUSH. This in turn should cause the disk to start and save the changes to Screen #72. Forth is not "dumb" like some other languages I know. It knows you want to save the corrections to the screen you were editing.

Now with that done, enter 72 EDIT as you did before and examine the screen. If you have an RS232 type printer, verify the Baud rate given on line 4. If the 600 is not correct change it by overtyping. If you need to insert something, press FCTN 2 (Insert), and Forth will make room for a character for each press. FCTN 1 (Delete) gets rid of an excess space or character. However, if you have a parallel printer you have more work to do. On lines 1, 2, 3 and 4 change RS232 to PIO, also on line 4 put anything else that may be required to describe your printer's characteristics. This may take some experimentation. When finished, press FCTN 9 (Back), then FLUSH to save the changes.

OTHER ERRORS

Additional errors have been found on the Forth System Disk. These are:

Screen #58, line 10 should be:

VDPMDE 4 < IF SMTN 80 VFILL 300 ' SATR ! ENDIF

The ! (store) as shown after 300 should ' (tick). Also correct line 1 of page 10 of Chapter 6 of the TIFM, to add ! (store) after SATR.

Screens #53,54,55, line 1, Change:

VDPSET2 to SETVDP2

Lastly on Screen #59,line 9 change:

OOFF to OOFE

I have not encountered any difficulty because of the above errors, probably because I have used sprites very little.

If you have trouble with this, please call me at, 714/847-5875 during computing hours. (normally 1:00 PM to 12:15 AM).

How do you know if you have trouble? First, type -PRINT (remember the MENU), wait for it to load and give back the cursor, then turn on the printer, or vice versa, then enter

72 SWCH LIST UNSWCH.

You should get a nice printout of Screen #72, if you do not, boy! (girl!) do you have trouble! Please go through the screen editing exercise again before you throw in the towel and call somebody. If you succeed with this then I suggest you print all the other screens you have changed and paste them in your TIFM.

If all this works as described you may start wearing your Forth Novice Badge.

Write protect your newly made and modified Forth Systems disk. We will be modifying it and the Editor later, but for now protect it. We are now ready to make a BSAVE'd Forth Working Disk. Make a newly formatted disk using Disk Manager or whatever. Then return to the E/A module and boot your new Forth System Disk. When you get the MENU, enter -64SUPPORT. After a while you will get the cursor back. Enter 72 EDIT, If you like what you see, then in the future when I refer to EDITOR, think 64SUPPORT. You cannot have both. I do not like it; I am quite ready to give up 64 column visibility for easy readability.

If you do not like 64 SUPPORT either, then type COLD. Forth should now reboot. This time select and enter -EDITOR (If you liked 64SUPPORT, then forget the COLD bit). Select and enter in turn -DUMP, -COPY, -PRINT, -BSAVE.

Now type VLIST, then press Space once. What you see before you is the Forth vocabulary you have just loaded. Gibberish? not quite, you will get used to it a little at a time. SPACE controls the scrolling and FCTN 4 (Clear) exits.

If you think you will be doing graphics, then also enter -GRAPH, -VDPMODES. If you think you MUST have floating point arithmetic, then also enter -FLOAT. The latter three utilities take up a large amount of memory, and if you do not use them, it is a waste. Floating Point is very slow compared to the rest of Forth. Type VLIST again to see what you have added to the dictionary. We will be covering Floating Point Arithmetic (FPA) and graphics later, but if you wish to barge ahead, do not let me discourage you, that is how I learned.

If you elected to load -GRAPH and -FLOAT, please enter FORGET COLTAB to unload them. Look with VLIST again, The first word should be BSAVE. If it is, good, else type -BSAVE again and after it loads, remove the Forth System Disk, insert your blank initialized disk, and type:

```
' TASK 20 BSAVE. ( do not forget the ')
```

This will save your dictionary in machine code starting on Screen #20 for fast loading on boot up, it should print 27 as the next available screen. Now while you have Forth loaded, reinstall your Forth System Disk in drive 2. If you only have a single drive, we have to do it a little differently, but I will explain that separately. Now type -

```
0 DISK_LO ! 180 DISK_HI !
```

Then 90 0 20 SMOVE to move Screens 0-19 from drive 2 to your new disk in drive 1.

If you do not have two disk drives, it is easier to start with a complete copy of your Forth System Disk to do your BSAVE to. This will just overwrite Screens 20 -26. The major disadvantage of this method is that you must then erase or over write all but the screens that are required for this version of Forth, namely; 0,1,3,4,5,8-26. This is not hard but sometimes it may be confusing as to what to keep and what to erase. The BSAVED screens look like garbage, do not erase them. To erase a screen, enter SCR# CLEAR FLUSH.

Next we will edit the Bootscreen #3. Enter 3 EDIT to get it on the CRT. Modify it to match the example Screen #3 below. Remember that FCTN 3 (Erase) will delete a line, and CTRL 8 will insert a blank line above the cursor. The revised Screen #3 should look like this.

```
     SCR #3
 0 ( WELCOME SCREEN)
 1 CLS 11 10  GOTOXY
 2 ." BOOTING  FORTH"
 3 HEX 10 83C2 C! ( Quit Off)
 4 DECIMAL 20 BLOAD 16 SYSTEM
 5   0 DISK_LO !
 6 180 DISK_HI ! ( for 2 SSSD drives)
 7 ( 90 DISK_HI !  for 1 SSSD drive )
 8 10 12 GOTOXY
 9 ." This Is My Very Own"
10 14 14 GOTOXY
11 ." Version Of"
12 10 16 GOTOXY
13 ." Fast Loading Forth"
14 1 23 GOTOXY
15
```

Actually lines 8-14 are just for fun, you do not need to do it. After you finish, do not forget FCTN 9 FLUSH. Now to test your handiwork do COLD. If all goes well you now have a customised Fast Loading Forth.

Now if you really need them, type -GRAPH, -VDPMODES, and/or -FLOAT again.

If you have not yet done so, I strongly recommend you try to find a first edition copy of STARTING FORTH by Leo Brodie. He can be fun to read even if you are not really interested in Forth. The cartoons alone are worth the money. Do not forget to read the TI Forth Manual Appendix C before you start reading STARTING FORTH.

If you are still with me to here, you may now change your badge to Forth Novice II.

May the FORTH be with you. o

Appreciate your Programmers

by Jim Peterson, Tigercub Software, USA

I quote from a letter recently received from Alexander Hulpke, the German student who wrote the TI-99/4A assembly version of Tetris which most of you have played -

"Many thanks for your letter and the offering to distribute Tetris. I encourage everyone to do so, if they do so at reasonable rates, as you do. It is a bit strange to see TexComp sell disks with my Tetris for $5 plus $3 shipping, which is more than I ask for the program."

"I do not think of the fairware as a way to earn money, the asked donation is merely for postage, sending updated versions, etc. It would be also great, if everyone who uses my programs would write to me, just to see the work is appreciated - but I think I do not have to tell you anything about this."

"Nevertheless, the response for Tetris was quite good, especially when thinking about the problems of sending (money) to Europe. Most people sent personal checks (which my bank will charge approximately $2.50 each when cashing several at once) as I found they did not trust the postal service to send cash, which is a bit easier."

Read that again - the part about "it would be great...if everyone...would write". I have had correspondence with many TI programmers. Some of them actually expected to make money by releasing programs as fairware - and they have almost invariably been disappointed. But, most of them want more than anything to know that someone is actually using the program which they worked so many hours to create!

Many of the programs being written nowadays for the TI-99/4A are extremely sophisticated. Those who have the skill and genius to write them could certainly better themselves if they abandoned us to write for computers which are still on the market and have an expanding user base (of users who are more accustomed to paying a decent price for software!).

So, you had better start appreciating what you are getting. If you find a program useful or enjoyable, whether it is fairware or public domain, if it has a programmer's name and address on it, spend five minutes of your time and a 25-cent stamp (It would be nice if that is all it cost here! ED) to tell him so! And if it is fairware, and you can possibly get a few grimy bills unstuck from your fingers, send them along.

continued on page 4

Spellbreaker part 2

This walk through is by Scorpia, Copyright 1984.

Ok, let's explore this new cube. Blorple "WATER" (guess where you are now?). You are about to take a dip in the ocean, and you do not want your spell book getting soggy, so put the book in the zipper, and close the zipper. Make sure you are holding the bread (this is another good place to save the game, just in case).

Go south. SPLASH! Oops, you dropped your "WATER" cube. And there is a grouper swimming nearby with a hungry look in its eyes. Drop the bread (so it will not eat the cube). As the grouper gobbles your offering, take the cube (before it sinks), then the bottle. Blorple "EARTH", then go east to the Hall of Stone (by the way, you will not need the "EARTH" cube again, so you can tuck that away in the zipper).

While you dry off, open the bottle and get the scroll that is inside (you can drop the bottle, which has no use in the game). This has the "Liskon" spell written on it. Get your spell book from the zipper, then Gnusto Liskon and learn it. Go north, and you come to a passage that is simply filled with a giant serpent (try Yomin on the snake some time).

Cast Liskon on the serpent, then continue north after it has shrunk a bit. Eventually you will come to the temple, which contains nothing of interest except a giant statue of a rat. If you climb up the statue and look in its mouth, you will see a cube sitting on the tongue. Unfortunately, there is no way for you to get your hand inside the mouth; the opening is just a bit too narrow.

What to do? Learn Malyon and Espnis. Save the game (because timing is very important here). Cast Malyon on the statue. The spell sure worked, because the statue is beginning to move. On the other hand, maybe it is not such a good thing, as the statue is looking around for something to nibble. Cast Espnis. This must be done at the right moment; if you wait too long, then when the rat becomes a statue again, it's mouth will not be open wide enough.

Ah, the rat is beginning to get sleepy, and just as it turns back into a statue, its mouth is wide open in a tremendous yawn. Just what you wanted! Climb up and get the cube from the mouth. Write "AIR" on the cube. Now, blorple "AIR".

Go north to the glacier (Brrr!) and get the scroll (which has the Tinsot spell). Be careful that you do not move at all while here, or you will have a very nasty accident! Blorple "AIR" again, then Gnusto Tinsot. Leave the room by going west.

Aha, you are in a bazaar! That shop with the carpets looks interesting; enter it by going east. The high-pressure salesman will show you two rugs; one is a beautiful blue carpet with a cube design, the other is a raggedy old thing of red material, suitable (maybe) for wrapping fish. ii has the blue one that you want here.

However, you can not just buy it; you will have to haggle with the merchant. Be careful to specify that it is the BLUE carpet you want. Now, when you try to buy it, the merchant will tell you it is 800 zorkmids. Not only is that an outrageous price, your gold coin is only good for 500 zorkmids. So you will have to get him to come down a bit.

Start by offering 100 zorkmids, and each round of haggling, raise your offer by 100 until he finally settles for 500. When you hand him the coin, he will give you a carpet. But wait!! Do not go running off just yet! The sneak has tried to pull a fast one on you.... he is given you the RED carpet! (The nerve of him!) So, before you leave, take the blue carpet. He will let you do that, but only if you have made sure to tell him that is the one you wanted during your haggling.

Now you can leave the shop and put the "AIR" cube in the zipper. Blorple "WATER". Learn the Tinsot spell at least 3 times. Put your book in the zipper again, save the game (in case of spell failure), then go north to the Oubliette.

Here you see a channel of water flowing between two pipes. Far above, you can also see a trap door. No ladder is handy, but there are other ways of getting up there. Tinsot the outflow pipe twice (once is not enough). As the room becomes filled with water, Tinsot the water. A small ice floe will form. Stand on that and wait until the water stops.

Now you can easily reach the trap door (Rezrov not necessary; you can open the trap by hand). Going up takes you to a slimy dungeon and another cube. Write "BONE" on this one and put it in the zipper; you do not really need to use it for anything (it takes you back to Belwit Square). Get your spell book, then saunter along east and north to the cell.

The furnishings of the cell have deteriorated with time, but that mahogany cabinet looks interesting. It has locked, so use Rezrov to open it. Well, looky there...a moldy old spell book was inside! Too bad you can not ready anything. But all is not lost! Learn the Caskly spell and cast it on the book. That is a little better, but not much. The only legible spell is the Snavig spell. Still, better than nothing, so Gnusto Snavig before the book falls apart.

Now, at last, it is time to deal with that pesky Roc (you have been waiting for this, have you not?). Go back to the dungeon, then up to the tower. There is a tiny dot approaching from above. This is the roc. Ignore it. Drop the carpet, sit on it, then say "fly". As you ascend from the tower, the Roc is so nonplussed it zooms off! (Now, why does that not work if you try it at the nest?)

With the Roc out of the way, the rest is simple. Just continue to fly west to the mountains and soon you will be above the nest. Go down until you land. Stand up, get the cube, then write "STRING" on it. Blorple "STRING". You will be in a room full of strands of light. Go south, and you are in the Enchanter's Retreat.

And here is Belboz. When you talk to him, you will find him a bit suspicious. Are you *really* you? You will have to prove it, by answering a question. This is random, and the answer can be found among the six cards that come with the game. Just read through them until you come to the right one, and give the name of the person on the card.

Having established your identity, you can now ask Belboz about the cubes. What he has to say is interesting, although perhaps not all that enlightening. However, he does give you a key, so hang on to it, because it is very important. You can also put the "STRING" cube into the zipper, as you will not be needing it again.

D/V 80 Files Converter

by Tom Wynne, USA

Tom Wynne, of the Puget Sound 99ers has developed a program called Pre-Formatter that may be of use to those who want to do global changes on TI-Writer files without using TI-Writer.

As it is listed here, the program will place carriage returns at the end of each line. It also gives the user the option of replacing all spaces with required space as symbolised by the caret. Users may easily modify this later option to permit search and replace of any character by changing line 200. In line 200, CI$ represents the character to be changes and C2$ represents the character you want to replace it with.

"I created it for the use of entering program listings and fixed format files in a document for use with the Formatter of TI-Writer", he wrote in his user group newsletter.

When the program is run, the user is prompted for the filename and drive designation of the file to be modified. If carriage returns already exist on the file, additional carriage returns will not be added.

The user should be able to modify the carriage return replacement operation as well be changing lines 360 and 370. The carriage return is identified by CHR$(13).

continued on page 22

Newsletter update

by Bob Relyea

Quite a few newsletters have piled up since we had our last update. As usual, I will give the national ones first and then the international ones.

HUNTER VALLEY 99ers, June, 1990: Brian Woods' US visit; various reviews such as 'Spell it', Tetris, Rock Runner, graphics utilities; Bob Carmany on TI Base and the "Do Set-Up" procedure; Call Key subroutines; prime number generator; Random Bytes by Bob Carmany, a program written in XB that converts between decimal, binary and hex; Far Out (in the bush) by Dick Schaydel; some Assembly language tips by Tony McGovern entitled 'Here There and Everywhere'; a transliteration file which helps draw up tables with TI-Writer; Nibblebytes by Al Lawrence; New Age 99 by Jack Sughrue.

October, 1990: A long review of the Quest RD200 Ramdisk by Bob Carmany which is manufactured by the Hunter Valley Users Group; a look at a couple of programs such as the German 'Mandelbrot' which deals with creating, saving and displaying fractal images as well as a program called 'Catlib' which is a type of disk catalog utility (review by Dick Schaydel); Random Bytes by Bob Carmany which deals with the use of the Quest Ramdisk; Bulletin Board Systems by Simon Phillips; New Age/99 Nos. 7 and 8 by Jack Sughrue; Data Communications, Chapter 2: Use of Smart/Automatic Modems by Geoff Phillips; latest Comprodine releases; Assembly Executing by Bob Carmany.

TIBUG (Brisbane), July, 1990: Ros in an Eprom (from TIsHUG); 'What's New - the latest on Rock-Runner and Press (Asgard is apparently selling the Beta Version with the binder, etc for $25); trading post; Expro Eprom Programmer Revisted by Col Christensen complete with two pages of schematics; Tips From the Tigercub #20; the winner of Micropendium's Contest (Col Christensen); review of 'Spell It'; Programme Listing - Ski Run; Disk Library; Saving Space by Garry Christensen.

August, 1990: Repairers Roundup by Col Christensen mainly involving transformers; Tips #21; Treasurers Report with graph drawn by TI-Tiler; Garry Christensen with an article on spelling and sounding words with he computer entitled 'Sounds Similar'; 'What's New' including a short mention of TIsHUG and our hardware projects; Programme Listing - Minefield; Hardware Report by Col Christensen - a rehash of the various things that have been in our magazine including 32K Console Memory Expansion and Console Tester; review of our magazine; Test Driving PC-Transfer by Ben Takach; computer dictionary; how to protect you programme from listing.

October, 1990: Module Order from Texcomp; financial report; 'What's New' GIFmania from Texaments, Hardback from T J Software, Golf Score Analyser by Harrison Software; newsletter update, international and local; Using 'Print Using' by Mark Schafer; Repairs and Ramdisks by Col Christensen; Beginners Stuff (mostly Print Using and Display); Sixteen Puzzle, August '83; Tips #23; How to use Multiplan from the Punn Newsletter; new commands with Advanced Basic demo (from Micropendium).

November, 1990: a memorial of a departed friend - Peter Glebe; What's New - new Myarc version of MDOS & Triad for the Geneve; Jack's Jottings by Jack Sughrue on 'Cassettes: the New Peripheral!'; Version 3.02 Extensions to TI-Base, Part I by Col Christensen; In The P.O. Box (Newsletter Digests); Tips No. 24 by Jim Peterson; The Canon Little Squirt BJ-10e portable printer; Oasis Pensive Abacuters, The Story Behind OPA by Garry Bowser; Fractal Explorer V.3 by Mark Schafer.

ATICC from Adelaide, November, 1990: Coordinator's Report; Editorial by Peter Giles; Dips and Tips by David Perkovic; ATICC Hall of Fame by David Perkovic; Extended Basic Tips No. 15 by Jim Swedlow; TI Bits No. 16 & 17 by Jim Swedlow; Memory Savers by Jim Swedlow; Data Writer V 1.2 by Jim Peterson; Tips No. 21 by Jim Peterson; Data Communications by Geoff Phillips; TI-Base Command File Utility by Bill Gaskill.

OVERSEAS NEWSLETTERS

SPIRIT of 99, Columbus, Ohio, May, 1990: What's Hot by Irwin Hott; Unlimited Basic by Jim Peterson; information on Star/Gemini Print Head Repair; Designing Characters Made Easy by Paul Scheidemantle; a page of 'tips' for the 99/4A; Root Finder published in the Ottawa User Group; Random Word Generator for adventure games; 3"x5" Card Catalog; TI-Base tutorials 14.1.1,2,3 by Martin Smoley.

June, 1990: 'What's Hott?' by Irwin Hott about uploading, mini-reviews of Animator Announcement, Air Taxi Review, Comic Show, Memorial Disk of Bill Knecht's music, TI-Base 3; Jim Peterson on 'Whatever Happened To Th Fun Of It?'; review of Air Taxi by Jim Peterson; Using a Modem -- Part 2 by Dick Beery; Tips #60 by Jim Peterson; New Age 99 #4 by Jack Sughrue; TI-Base Tutorials 14.1.1,2 by Martin Smoley.

July, 1990: Using a Modem -- Part 3 by Dick Beery; What's Hott with reviews of Diskread, Precise Math by Jim Peterson, Patch97H Rev, Basic Compiler, Triad, Applejam; New Age/99 #5 by Jack Sughrue; The TI Learning Machine by M. Amundsen; Kids Korner with Fairy Tales by Chick De Marti; TI-Writer File Keeping by Paul E. Scheidemantle (good suggestions on how to set up disks; TI-Base Tutorials 15.1.1,2 by Martin Smoley.

October, 1990: What's Hott with an up-date on the Clearinghouse BBS idea and reviews on Letters to Santa Claus, Shopping Lister, Yapp Announcement; Tips # 60 by Jim Peterson; a review of Mass Transfer by Ted Stringfellow; TI World News by Jim Peterson with a mention of meetings and announcements; What Do I Have To Do Anyway? by Jim Peterson; A review of Page Pro by Jack Sughrue in New Age/99; TI-Base Tutorials 17.1.1,2 by Martin Smoley.

November, 1990: TI's in Europe, a report from Germany; What's Hott-the Chicago Faire as well as several software reviews; The Home Computer by Jim Peterson; TI World News by Jim Peterson; Language Drills by Don Shorock, a review by Jim Peterson; Holiday Tips by Ed Machonis; New Age/99 No. 8 by Jack Sughrue.

TOPICS-LA 99ers, August, 1990: Word Fun No.2 (with answers) by Earl Raguse; Continuing Forth #1 &2 by Earl Raguse; The Cracker Barrel by Chick De Marti with a TI-Writer Tip, a Forth Tip, XBasic Tip, Call Init, A TI Vacation Anybody?; Part 7 of a list of items in the software library; a mention of Funnelweb V.4.21, Multiplan Exercises by Herbert Sclesinger, TI Yatzi, Contract Bridge by John Bull and 'Gaskill Stuff'; a list of club membership.

October, 1990: Using the 2 column program by Earl Raguse (long article); 'To My Darling Husband' (for the computer widow!); TI-Multiplan V4.0 a review by Audrey Bucher; a list of Forth and XBasic Similarities; The Cracker Barrel by Chick De Marti including 'Protect Your Program' by Jim Peterson, RAG Software and the Multiplan V4.0, Troubleshooting Checklist, TI-Base V3.0, EZ - Programming, TI-Base Printer Codes; a list of Asgard Software; TI-Base to TI-Writer by Fred Moore.

November, 1990: Thoughts from the President; Word Fun No. 3 by Earl Raguse; Subprograms and How They Work by Earl Raguse; Ordering back issues of 'Topics'; The Cracker Barrel by Chick De Marti; Extended Basic Tips No. 15 by Jim Swedlow; A Child's First Look at Multiplication by Tony Falco; NASA Frequencies, edited by Henry Badon; a list of recent LA Software as well as items from the Fairware library.

The PUG PERIPHERAL, May, 1990: Bulletin Board News by Gary Taylor; a review by Sue Harper of Jim Peterson's Homonymy; Kiddies Corner by Sue Harper with a few nice, short programs; From The Librarian; New Age/99 #4 by Jack Sughrue with a list and short description of new software, mostly games; Tips #51 by Jim Peterson; TI-Multiplan V4.0, a review by Audrey Bucher.

July, 1990: On BBS by Gary Taylor; In The Spirit of TI by Christopher Pratt (Spring 1990 "All Electronics Corp" catalog list TI products such as power supplies for the 99/4A); Page-Pro/99 - Part 1 by Jack Sughrue; 'My Loaders' by Rick Keppler or How to write your own quick-and-dirty menu-driven X/B loaders that load darn-near anything!.

October, 1990: CC-40 and Stuff (including the pin diagrams) by Gary Taylor; Exclaiming GIF graphics on the TI by Christopher Pratt; Telco Curiosity by Gary Taylor; Dividing Five Databases Into Ten In TIBase by Lynn Gardner; New Age/99 #8 by Jack Sughrue about newsletters; The Adventurous Computerist by Rick Link; Printers No. 9 by John Willforth including diagrams.

November, 1990: New Age/99 No. 9 by Jack Sughrue; The Asgard Mouse; a list of text adventure games available for the TI-99/4A by Rich Link; TI-Base report TIP/BUG; the Kiddie Corner by Sue Harper; Tips No. 53 by Jim Peterson.

TIDBITS, July, 1990: In The News by Gary Cox, possibilities of "speech modules for the 99/4A, another new Grom device and a TI card for IBM compatibles, FAX machines available for the TI, a new generation Geneve may be in the making, Press is 'on the ropes', Triton has dropped two of its catalogs; Asgard's - The Animator by Brad Ryder; Genial TRAVelER (from June,1990 Micropendium); various advertisements; Speeding Up TI-Base by Bill Gaskill.

August, 1990: In The News - where to buy video modulators, the Chicago TI International World's Faire, the new BBS of the Phoenix, Arizona; The Mystery of Line 32767 by Andy Frueh; Surge Protectors by Andy Baird; Surge Protectors by Bill Gaskill.

October, 1990: In The News - Chicago's TI World Faire, Texaments GIFmania, various edited bit and pieces from September, 1990 Micropendium; No Mystery About line 32767 by Andy Frueh; Appreciate Your Programmers by Jim Peterson; Language Drills by Don Shorock; Tips No. 17 by Jim Peterson; Notes on German TI Faire.

November, 1990: President's Bit; In The News by Gary Cox - a word about Terry Miller's Triton Products; The Chicago Faire Report by Gary Cox; The OPA Oasis Pensive Abacuters TI-Image-Maker; Wiesbaden TI-Fair by Beery Miller; a list of TI suppliers by Gary Cox.

UGOC ROM, August, 1990: President's Column; UGOC Hall Of Fame by Bill Nelson and Siles Bazerman; And So Forth by Earl Raguse.

October, 1990: President's Column by Ben Hatheway; UGOC Hall Of Fame by Bill Nelson; Graphic Sig by Bill Nelson; Dips And Chips by Siles Bazerman; Newsletter Tidbits No. 14 by Earl Raguse; TI Bits No. 22 by Jim Swedlow including 'Incorporating Instances' and 'Mail Merge'; advertisement of Faire West '91; Notes on the Zeno Board by JFW with diagrams.

November, 1990: From the President by Ben Hatheway; Graphic SIG; Membership Corner by Bill Nelson; In MY Humble Opinion by Bill Nelson - mostly about cataloguing; Dips and Chips by Siles Bazerman; Tips or TI Print Shop by Earl Raguse (XBasic tips, TIPSLABEL & TIPSSHOW); How to program 25 series eproms when your programmer only does 27xx by Dan Eicher together with the necessary diagrams; Fest West 1991 Update.

TI Focus, October, 1990: News and Views by Tom Arnold including MMXB or Multimode Extended Basic by Jean Marleau; Club Page by Tor Hansen - perhaps a new 80 column card to take up where Digit left off, various reviews of software such as mentioned above in the other magazines; Appreciate Your Programmers by Jim Peterson; More TI-Base by Rick Lilley (long article); Toshiba T1200XE by Tom Arnold; New Age/99 No. 8 by Jack Sughrue.

November, 1990: News and Views by Tom Arnold; 9600 Baud Modems Bargain Price by John Van Weelie - CH99; Whoops! by Rick Lilley; Whys I Love My Computer, BUT Not Necessarily My Users Group! by Randy Packham; Club Page by Tor Hansen; Chicago TI Faire 99 by John Van Weelie; a review of Jim Peterson's public domain library collection - disk 900 by Tom Arnold; Batch-It by Ron Marissen; Doctoring Your Disks Part II by Dwayne Verhey; New Age/99 No. 9 by Jack Sughrue.

PUNN NEWSLETTER, September, 1990: Gemini Control Codes; Kidstories - Number Game; Tips v.1.3, an extensive free graphics program by Eugene Smith; Letterform by Ollie Herbert; Symbolism by Stan Corbin; Read the Docs, by Stan Corbin; Games to type in entitled Fortunes by Ed Machonis; Form Letters With TI-Writer; Word Safari program by Jack Sughrue; Addressing Envelopes With A Gemini 10X or SG10 Printer by Charles Good; Computer Keeps Car On Straight and Narrow by Joseph Szczesny.

BOSTON COMPUTER SOCIETY, June/July, 1990: Listen by Justin Dowling; Introduction To The UCSD P-System by Ron Williams; Using Multiplan by D.L. Mahler; A Multiplan Tip by Justin Dowling; R.L.E. Digitizer! by Ray Kazmer; Far Out by Dick Schaydel; Beating Around The Bush by Ron Klienschafer.

August/September, 1990: LIsten by Justin Dowling; Using Multiplan by D.L Mahler (Parts 1 and 2 of Multiplan and Taxes); Wizziwig, a review of Page Pro 99 by Justin Dowling.

October, 1990: Listen by Justin Dowling; Notes On The German TI Fair by Beery Miller; Appreciate Your Programmers by Jim Peterson; Disk Drive specifications by Louis Guion; Self Diagnosis (origin unknown).

LEHIGH 99'ER COMPUTER GROUP, August, 1990: President's and Editor's Column; Ripples Revisted; Researchers are Boolish Too! by Carney Mimms; Taking the Bool By The Horns (or the practicalities of programming using Boolean Algebra) by Art Byers; Genealogy Programs for the TI 99/4A Computer by John Geisinger; New Age 99 No. 8 by Jack Sughrue; Cemetery Hoppin, compiled by John Geisinger from the notes of Nelson Eddy (wonder if he can sing?).

CIM 99 (Montreal), October, 1990: Assembly Made Easy by Daniel Lauzon; TI-Base by Andre St-Aubin; Programmes on the Disk of the Month; Swan Graphic Writer version 3.0 by Sylvain Mornard; a list of library software (N.B. most of the magazine is in French).

continued from page 20

Wynne warns that large files will result in a "memory full" error message.

Editor's Note - I have keyed the program in (below) and used it so I can assure you that it is correct. I listed it to disk from a working program to create the DV 80 file that was used to write this article.

```
100 !**********************
110 !*   PRE-FORMATTER    *
120 !* PUTS CR's AT END   *
130 !* OF EACH LINE AND   *
140 !*  REPLACES SPACES   *
150 !*   WITH REQUIRED    *
160 !*       SPACE        *
170 !*   BY TOM WYNNE     *
180 !**********************
190 DIM A$(300)
200 C1$=" " :: C2$="^"
210 PRINT "ENTER FILE NAME: "
220 ACCEPT BEEP:FN$
230 PRINT "REPLACE ";C1$;" WITH ";C2$;"?";
240 ACCEPT VALIDATE("YN"):YN$ :: IF YN$="" THEN 240
250 OPEN #1:"DSK2."&FN$,INPUT
260 I=0
270 PRINT "READING FILE..."
280 IF EOF(1)THEN 320
290 LINPUT #1:B$
300 IF YN$="Y" THEN CALL REPLACE(B$,C1$,C2$)
310 A$(I)=B$ :: I=I+1 :: GOTO 280
320 CLOSE #1
330 PRINT "WRITING FILE..."
340 OPEN #1:"DSK2."&FN$,OUTPUT
350 FOR J=0 TO I-1 :: IF A$(J)="" THEN 370
360 IF SEG$(A$(J),LEN(A$(J)),1)=CHR$(13)THEN 380
370 A$(J)=A$(J)&CHR$(13)
380 PRINT #1:A$(J):: NEXT J :: CLOSE #1
390 PRINT "FINISHED."
400 GOTO 210
410 SUB REPLACE(A$,C1$,C2$)
420 B$=""
430 FOR I=1 TO LEN(A$)
440 CH$=SEG$(A$,I,LEN(C1$))
450 IF CH$=C1$ THEN CH$=C2$
460 B$=B$&CH$ :: NEXT I :: A$=B$
470 SUBEND
```

Regional Group Reports

Meeting Summary For FEBRUARY

Banana Coast	10/02/91	Sawtell
Carlingford	20/02/91	Carlingford
Central Coast	09/02/91	Saratoga
Glebe	07/02/91	Glebe
Illawarra	11/02/91	Keiraville
Liverpool	08/02/91	
Northern Suburbs	28/02/91	
Sutherland	15/02/91	Jannali

BANANA COAST Regional Group
(Coffs Harbour area)

Regular meetings are held in the Sawtell Tennis Club on the second Sunday of the month at 2 pm sharp. For information on meetings of the Banana Coast group, contact Kevin Cox at 7 Dewing Close, Bayldon, telephone (066)53 2649, or John Ryan of Mullaway via the BBS, user name SARA, or telephone (066)54 1451.

CARLINGFORD Regional Group

Regular meetings are normally on the third Wednesday of each month at 7.30pm. Contact Chris Buttner, 79 Jenkins Rd, Carlingford, (02)871 7753, for more information.

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043)69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce St, Glebe. Contact Mike Slattery, (02)692 0559.

ILLAWARRA Regional Group

Regular meetings are normally on the second Monday of each month, except January, at 7.30pm, Keiraville Public School, Gipps Rd, Keiraville, opposite the Keiraville shopping centre. A variety of activities accompany our meetings. In the December meeting Geoff gave a demonstration of the new Funnelweb from Tony McGovern and Bob ran through the 'Missing Link' demonstration disk. Contact Lou Amadio on (042)28 4906 for more information.

LIVERPOOL Regional Group*

Regular meeting date is the Friday following the TIshug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 6447377 (home) or (02) 7598441 (work) for more information.

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02)918 8132.

Come and join in our fun. Dick Warburton.

SUTHERLAND REGIONAL GROUP

All meetings are held on the third Friday of each month at 51 Jannali Avenue, Jannali unless otherwise advised.

Peter Young
Regional Co-ordinator

TIsHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

FEBRUARY MEETING - 2ND FEBRUARY

The February meeting will be the first meeting of the year and unlike previous years it will not start with the AGM. The AGM was held in December so we can start off the new year getting right into the computer. Much new software has been acquired from the U.S. such as The Missing Link and there is a distinct possibility that some of this will be demonstrated at the meeting. There is also talk at this stage of the March meeting being a Buy, Swap and Sell day.

****

The cut-off dates for submitting articles to the Editor for the TND are:

March	10 February
April	10 March

****

TIsHUG Meetings for Sydney, 1991

February
The first meeting of the year starts with two demonstrations. The first of these is a RAMdisk with twelve 32Kbyte static RAM chips and ten 64Kbyte EPROM chips to make about 1Mbyte of storage in two layers of chips. Lou will demonstrate how much software can be accessed quickly and easily using the device. The second demonstration will be Spell It!, the new spelling checker from Asgard. This is a very useful utility program for word processing.

March
The club has the excellent music software from Harrison Music for sale and the demonstration this month will be the MIDI music interface. This is an industry standard, and should enhance the already excellent music capabilities of our computer.

April
This is the first of the Buy, Swap and Sell days. These will all be held on the meeting days which fall in the school holidays. Mark these down on your calendar and make sure that you get to at least one of them.

May
The first all day tutorial session. Get in early and tell Russell what you would like a tutorial on. We can arrange for tutorials on any subjects as long as you give us some warning.

June
Demonstrations of the latest software or hardware. Watch this space for details.

July
The second Buy, Swap and Sell day. Bring things for sale and also your money for the terrific bargains. You have to be early to beat Rolf to a bargain!

August
Demonstrations of the latest software or hardware. Watch this space for details.

September
Demonstrations of the latest software or hardware. Watch this space for details.

October
The third Buy, Swap and Sell day. Your last chance this year to get some money for Christmas presents or to get an early present for yourself.

November
The second all day tutorial session.

December
The Annual General Meeting followed by some festive eats and drinks. Make sure you attend and give your support to all the workers in the club.